国外电子与通信教材系列



CMOS 集成电路版图

——概念、方法与工具

CMOS IC Layout

Concepts, Methodologies, and Tools



[加] Dan Clein 著

邓红辉 王晓蕾 耿罗锋 等译

高明伦 审校



電子工業出版社
Publishing House of Electronics Industry
http://www.phei.com.cn

CMOS集成电路版图

——概念、方法与工具

- 学习VLSI设计的学生需要有一本针对IC版图设计的教科书
- 全球的集成电路版图设计人员需要一门详细的培训课程
- 设计工程师需要一本可以帮助他们理解版图设计概念的书，以使他们的设计更接近现实
- 软件开发人员需要更广泛的信息来源
- 市场营销人员需要一本专用词典去了解客户需要的术语和概念
- 现在你手上有了一把钥匙——《CMOS集成电路版图——概念、方法与工具》

目前，随着新的工具、平台、方法和设计方式不断涌现出来，设计工程师需要一本涉及尖端技术的完整的参考手册和培训指南。大部分关于VLSI的教科书涵盖了电路设计技术或者CAD工具的算法，但是没有一本书广泛地解释了IC版图设计所面临的挑战。本书不仅阐述了IC版图设计的基础知识和理论，也提供了一个有用的循序渐进的过程，使得每个设计工程师的工作变得更轻松、更有效。

本书基于作者的工作经验提供技能训练，给出了新的过程和设计指南。本书作者开发过高级版图设计程序，改进过设计方法和设计流程。他使用过版图规划工具、标准单元、布局布线工具、高级模拟版图设计工具，以及非常有效的层次化验证方法。他的设计团队在生产率上实现了200%的提高，你也可以做到！使用书中阐述的概念和工具，你可以提高你的工作技能。



本书附带光盘包含有全面的集成电路版图和Tanner软件演示版

作者简介

Dan Clein 是MOSAID技术公司半导体分部IC版图设计经理，他为设计人员做的内部培训是加拿大这类教程中仅有的。

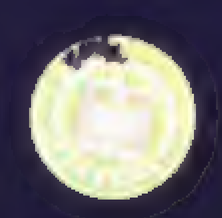


ISBN 7-121-02303-2

9 787121 023033 >

本书译自原版 CMOS IC Layout: Concepts, Methodologies, and Tools，并由Elsevier授权出版。







限在中国大陆地区销售





责任编辑：赵红燕
责任美编：毛惠庚

ISBN 7-121-02303-2　　定价：29.00元（附光盘1张）

国外电子与通信教材系列

CMOS 集成电路版图

——概念、方法与工具

CMOS IC Layout

Concepts, Methodologies, and Tools

[加] Dan Clein 著

邓红辉 王晓蕾 耿罗锋 等译

高明伦 审校

電子工業出版社

Publishing House of Electronics Industry

北京 · BEIJING

内容简介

本书以循序渐进、深入浅出的方式，系统地介绍了CMOS集成电路版图设计的基本概念、设计理念和各种方法技巧。全书共分10章，阐述了版图设计技术的基本概念和设计理念，当今流行的几种基本设计流程，专用模块的版图设计技巧，版图设计的高级技术和深层次概念，版图设计的基本工具类型，工具的特性和典型用法。与其他IC设计教程相比，本书注重理论与工程实践的结合，书中提供了大量实例来帮助读者正确理解版图设计的基本概念和关键设计理念，生动形象，简明易懂，可读性强。

无论对版图设计工程师，还是对电路设计工程师、CAD人员、学习IC设计的学生，本书都是一本非常不错的参考指南和培训教程。

图书在版编目（CIP）数据

CMOS集成电路版图——概念、方法与工具/（加）格雷（Clein, D.）著；邓红辉等译.
北京：电子工业出版社，2006.3
（国外电子与通信教材系列）
书名原文：CMOS IC Layout: Concepts, Methodologies, and Tools
ISBN 7-121-02303-2

Ⅰ.C… Ⅱ.①格… ②邓… Ⅲ.集成电路-设计-教材 Ⅳ.TN402

中国版本图书馆CIP数据核字（2006）第012210号

责任编辑：赵红燕　　特约编辑：詹文军
印　　刷：北京市顺义兴华印刷厂
出版发行：电子工业出版社
　　　　　北京市海淀区万寿路173信箱　邮编：100036
经　　销：各地新华书店
开　　本：787×980　1/16　印张：16.25　字数：373千字
印　　次：2006年3月第1次印刷
定　　价：29.00元（附光盘1张）

序

2001年7月间，电子工业出版社的领导同志邀请各高校十几位通信领域方面的老师，商量引进国外教材问题。与会同志对出版社提出的计划十分赞同，大家认为，这对我国通信事业、特别是对高等院校通信学科的教学工作会很有好处。

教材建设是高校教学建设的主要内容之一。编写、出版一本好的教材，意味着开设了一门好的课程，甚至可能预示着一个崭新学科的诞生。20世纪40年代MIT林肯实验室出版的一套28本雷达丛书，对近代电子学科、特别是对雷达技术的推动作用，就是一个很好的例子。

我国领导部门对教材建设一直非常重视。20世纪80年代，在原教委教材编审委员会的领导下，汇集了高等院校几百位富有教学经验的专家，编写、出版了一大批教材；很多院校还根据学校的特点和需要，陆续编写了大量的讲义和参考书。这些教材对高校的教学工作发挥了极好的作用。近年来，随着教学改革不断深入和科学技术的飞速进步，有的教材内容已比较陈旧、落后，难以适应教学的要求，特别是在电子学和通信技术发展神速、可以讲是日新月异的今天，如何适应这种情况，更是一个必须认真考虑的问题。解决这个问题，除了依靠高校的老师和专家撰写新的符合要求的教科书外，引进和出版一些国外优秀电子与通信教材，尤其是有选择地引进一批英文原版教材，是会有好处的。

一年多来，电子工业出版社为此做了很多工作。他们成立了一个“国外电子与通信教材系列”项目组，选派了富有经验的业务骨干负责有关工作，收集了230余种通信教材和参考书的详细资料，调来了100余种原版教材样书，依靠由20余位专家组成的出版委员会，从中精选了40多种，内容丰富，覆盖了电路理论与应用、信号与系统、数字信号处理、微电子、通信系统、电磁场与微波等方面，既可作为通信专业本科生和研究生的教学用书，也可作为有关专业人员的参考材料。此外，这批教材，有的翻译为中文，还有部分教材直接影印出版，以供教师用英语直接授课。希望这些教材的引进和出版对高校通信教学和教材改革能起一定作用。

在这里，我还要感谢参加工作的各位教授、专家、老师与参加翻译、编辑和出版的同志们。各位专家认真负责、严谨细致、不辞辛劳、不怕琐碎和精益求精的态度，充分体现了中国教育工作者和出版工作者的良好美德。

随着我国经济建设的发展和科学技术的不断进步，对高校教学工作会不断提出新的要求和希望。我想，无论如何，要做好引进国外教材的工作，一定要联系我国的实际。教材和学术专著不同，既要注意科学性、学术性，也要重视可读性，要深入浅出，便于读者自学；引进的教材要适应高校教学改革的需要，针对目前一些教材内容较为陈旧的问题，有目的地引进一些先进的和正在发展中的交叉学科的参考书；要与国内出版的教材相配套，安排好出版英文原版教材和翻译教材的比例。我们努力使这套教材能尽量满足上述要求，希望它们能放在学生们的课桌上，发挥一定的作用。

最后，预祝“国外电子与通信教材系列”项目取得成功，为我国电子与通信教学和通信产业的发展培土施肥。也恳切希望读者能对这些书籍的不足之处、特别是翻译中存在的问题，提出意见和建议，以便再版时更正。

吴佑寿

中国工程院院士、清华大学教授

“国外电子与通信教材系列”出版委员会主任

出 版 说 明

进入21世纪以来，我国信息产业在生产和科研方面都大大加快了发展速度，并已成为国民经济发展的支柱产业之一。但是，与世界上其他信息产业发达的国家相比，我国在技术开发、教育培训等方面都还存在着较大的差距。特别是在加入WTO后的今天，我国信息产业面临着国外竞争对手的严峻挑战。

作为我国信息产业的专业科技出版社，我们始终关注着全球电子信息技术的发展方向，始终把引进国外优秀电子与通信信息技术教材和专业书籍放在我们工作的重要位置上。在2000年至2001年间，我社先后从世界著名出版公司引进出版了40余种教材，形成了一套"国外计算机科学教材系列"，在全国高校以及科研部门中受到了欢迎和好评，得到了计算机领域的广大教师与科研工作者的充分肯定。

引进和出版一些国外优秀电子与通信教材，尤其是有选择地引进一批英文原版教材，将有助于我国信息产业培养具有国际竞争能力的技术人才，也将有助于我国国内在电子与通信教学工作中掌握和跟踪国际发展水平。根据国内信息产业的现状、教育部《关于"十五"期间普通高等教育教材建设与改革的意见》的指示精神以及高等院校老师们反映的各种意见，我们决定引进"国外电子与通信教材系列"，并随后开展了大量准备工作。此次引进的国外电子与通信教材均来自国际著名出版商，其中影印教材约占一半。教材内容涉及的学科方向包括电路理论与应用、信号与系统、数字信号处理、微电子、通信系统、电磁场与微波等，其中既有本科专业课程教材，也有研究生课程教材，以适应不同院系、不同专业、不同层次的师生对教材的需求，广大师生可自由选择和自由组合使用。我们还将与国外出版商一起，陆续推出一些教材的教学支持资料，为授课教师提供帮助。

此外，"国外电子与通信教材系列"的引进和出版工作得到了教育部高等教育司的大力支持和帮助，其中的部分引进教材已通过"教育部高等学校电子信息科学与工程类专业教学指导委员会"的审核，并得到教育部高等教育司的批准，纳入了"教育部高等教育司推荐——国外优秀信息科学与技术系列教学用书"。

为做好该系列教材的翻译工作，我们聘请了清华大学、北京大学、北京邮电大学、南京邮电大学、东南大学、西安交通大学、天津大学、西安电子科技大学、电子科技大学、中山大学、哈尔滨工业大学、西南交通大学等著名高校的教授和骨干教师参与教材的翻译和审校工作。许多教授在国内电子与通信专业领域享有较高的声望，具有丰富的教学经验，他们的渊博学识从根本上保证了教材的翻译质量和专业学术方面的严格与准确。我们在此对他们的辛勤工作与贡献表示衷心的感谢。此外，对于编辑的选择，我们达到了专业对口；对于从英文原书中发现的错误，我们通过与作者联络、从网上下载勘误表等方式，逐一进行了修订；同时，我们对审校、排版、印制质量进行了严格把关。

今后，我们将进一步加强同各高校教师的密切关系，努力引进更多的国外优秀教材和教学参考书，为我国电子与通信教材达到世界先进水平而努力。由于我们对国内外电子与通信教育的发展仍存在一些认识上的不足，在选题、翻译、出版等方面的工作中还有许多需要改进的地方，恳请广大师生和读者提出批评及建议。

电子工业出版社

教材出版委员会

译 者 序

无论数字集成电路设计还是模拟/混合集成电路设计，版图设计都是必不可少的重要设计环节。随着芯片规模的不断增长，工艺的不断进步，设计复杂度的不断提高，版图设计越来越凸显出它的重要性并逐渐成为IC设计的一个新兴领域，它直接影响着芯片流片是否成功，芯片性能的好坏，芯片的成本以及面市时间。

我国的集成电路（IC）产业近年来发展非常迅速，对IC设计人才的需求日趋膨胀，尤其在版图设计方面的人才更是紧缺，因此，版图设计人才的培养就变得越来越迫切。然而，国内这方面的教材和科技读物屈指可数，这给广大有志于从事版图设计工作的读者带来了诸多不便。我们很高兴能与电子工业出版社合作，翻译出版《CMOS集成电路版图——概念、方法与工具》这本国外IC版图设计的经典教材，希望能够对广大读者有所帮助。

本书作者Dan Clein是MOSAID技术有限公司半导体分部的IC版图设计经理，也是一位资深设计工程师。本书将他十几年来积累的丰富设计经验浓缩于其中，以循序渐进、深入浅出的方式，系统地介绍了CMOS集成电路版图设计的基本概念、设计理念和各种方法技巧。

与其他IC设计教程相比，本书覆盖面非常广，不仅系统地阐述了版图设计的基本概念和相关技术，还讨论了IC版图设计所面临的挑战。本书也注重理论与工程实践的结合，书中提供了大量实例来帮助读者正确理解版图设计的概念和关键设计理念，生动形象、简明易懂、可读性强。

无论对版图设计工程师，还是对电路设计工程师、CAD人员、学习IC设计的学生，本书都是一本非常不错的参考指南和培训教程。

本书的翻译工作由合肥工业大学的高明伦教授主持，邓红辉翻译了前言、第2到4章、第10章和附录，王晓蕾翻译了第5章和第8章，耿罗锋翻译了第1章和第7章，刘聪翻译了第6章，章伟翻译了第9章。全书由高明伦教授审校。在翻译本书的过程中，孙华波、圣应山、张俊、许海辉、于冰、何伟、侯宁、何亚军、瞿美霞、梁上泉、张文婷、沈斌等给予了大力支持。另外，黄敏和李慧娟参加了本书的校对工作，在此向他们表示感谢。最后感谢潘剑宏副教授、王锐老师、林微老师的帮助和支持，感谢合肥工业大学微电子设计研究所全体师生的关心和支持。

由于时间仓促，加上译者水平有限，书中难免有错误和不妥之处，敬请广大读者批评指正。

前　言

从前，大约是1988年，在摩托罗拉半导体以色列公司（MSIL）成功完成一项颇有压力而又十分成功的项目后，整个团队被邀请参加一个特别的午餐会。所有人都为提前完成项目感到高兴，聚在一起欢庆交付流片的成功。和平时分开坐不同，这一次IC电路设计工程师、CAD技术支持人员以及IC版图设计工程师都围着圆桌坐在一起。我有幸坐在当时公司的首席执行官Zvi Soha旁边。饭后，在上甜点之前，Zvi问我们，怎样才能使每一个人都心情愉快，效率更高，也就是说，怎样提高生产率。一时间，众说纷纭：

IC设计工程师要求有更快的工作站、更多的仿真软件和更多的工程师。

IC版图设计工程师要求有更快的机器和布局布线工具、更多的人手和CAD组更多的支持。

CAD组的代表说他们需要的是更多的人手，因为他们想为摩托罗拉公司提供一套完整的软件解决方案，利用这套软件，首席执行官只要按一个键，一个完整的芯片就能在瞬间准备好。也就是说，如果Zvi需要一款新的芯片，这套软件就会弹出一张表，他只需要在表上的相应栏中填上特定参数，然后按回车键，就能得到最终设计。这个CAD代表继续解释说："有了这种功能强大的软件，就不需要这些总是要求更多软件和更快硬件的设计工程师和版图设计工程师了。"

几分钟之后，Zvi的回答是："很好，但如果有功能这么强大的软件的话，我也不需要你们CAD组了……"

这个现实生活中的故事的寓意就是，在10年前，大多数人普遍认为，使用先进和高尖端的软件就能解决所有的主要问题。

一个不可否认的事实是，随着器件的栅长变得越来越小，芯片的集成度逐渐增加，设计复杂度也逐渐增加，而产品的面市时间则要求更短，因此，设计团队必须寻求一种新方法来应对这些挑战。

对于设计自动化人员来说，他们难以理解的是，当一种新的自动化设计工具被普遍采用时，原有的挑战就已经发生了变化。

例如，当模块的大小和设计复杂度增长到靠手工难以完成其版图设计时，版图规划工具和布局布线工具就被引入进来，从而使版图设计过程自动化。

开始的时候，这些工具都是由基于电路图的设计方式驱动的。但是随着电路的复杂度和规模不断增长，就开始采用CAD，并且出现了综合工具。

下一步是采用布局布线工具等进行综合。如果我们分析所有自动化软件的发展，可以发现，所有的发展都是由那些做好充足准备、深知事物发展方向、有能力为新问题寻求新解决方

案并进行变革的人驱动的。驱动所有工具发展的是那些为改变做好准备的人，而不是那些通晓事物的原理并且有能力为新问题寻求新解决方案并进行变革的人。

诚然，自动化有助于改进设计——但是改变和改进设计的人，往往是那些懂得基础概念、尝试新方法的人，而且他们还推动着CAD软件设计者们继续开发新软件。

正是出于这个原因，我将帮助所有感兴趣的设计者，无论电路设计工程师还是版图设计工程师，以及CAD开发人员，了解更多版图的真实情形。这也就是本书之所以花大量篇幅讲述概念、方法学以及与CMOS版图设计有关的工具的原因。

几年前，在设计自动化会议上，我被邀请观看了一种新的布局布线工具的演示。在试映厅进行的10分钟的演示过程中，该工具的卓越性能给我留下了深刻的印象，所以私下我又要求观看了一遍40~50分钟的演示。

在试映厅里，有5个来自不同公司的人。那个软件开发者，对他的性能卓越的工具十分自豪，他开始解释有关工具的所有特性。大概过了30分钟，他开始在不同层次的集成环境上演示许多充满版图规划选项的屏幕镜头，这使我们所有人都大吃一惊。这个功能强大的工具给每个人都留下了深刻的印象。

在最后5分钟，这个软件开发者试图请我们这些潜在的软件用户提些问题。屋子里很安静……在问了一个非常老套的问题之后，每个人都快速地离开了。

当我和这个软件开发者单独在一起的时候，我问了他一些问题，记录如下：

在开发这个工具的过程中，是否有人考虑过潜在的用户——他们是谁？他们的软件知识水平怎样？从他们需要设置的参数的数量来看，使用这种工具并不是一件很容易的事。假设是软件背景知识有限的人使用这个工具，可能有200个以上的参量需要手工完成，其他一些参量则会自动设置。只有如此之后，才能按一下按钮得到理想的结果。可如果设计需要更多的调整，其用户可能只有向专家咨询，或者尝试掌握该工具的高级特性和功能。

回答是，“我们没有考虑过这个问题……”

这种工具的推销倘若仅重视宣传该工具的高级功能，是远远不够的。使用的简易性是一个非常关键却容易被忽视的问题！

我曾建议开发团队设一个顾问委员会，其成员应由来自不同公司的潜在用户组成，他们会采用不同的方法，会有不同的需求。我问，你们有这样的顾问委员会吗？

在问过几个类似的问题之后，我意识到这个由20名软件工程学博士组成的开发小组，只具有非常有限的有关物理版图的知识和经验，他们是基于规范开发出这个功能强大的工具的，完全没有和潜在的用户沟通过，也没有得到任何的反馈。

这就是我在构思本书的时候所思考的一些问题：如果不了解有关版图的概念和方法，要设计和构建出一个版图设计工具是非常困难的。

正是基于以上这些原因，我非常遗憾地说，这个“很棒的”工具至今尚未投放市场，用户当然也无从受益（很抱歉，这里我们省去该公司的名字）。

这些年来，类似的事情发生过很多次，所以，我决定对这些工具开发人员伸出援手。的确，我们需要更好的工具，但作为用户我们也必须帮助工具开发者更多地了解工具所涉及的领域内的基本原理。同时，也必须对工具的基本原理有更多的了解。一旦决定开发一项工具，负责制定规范的技术市场部必须明确一些东西，并使之得以体现在最终的工具中。

作为用户，使用新的工具意味着我们必须调整自己的思路和方法以适应新工具。关于这一点，专用集成电路（ASIC）的设计流程就是最好的例子。由于这种流程上的新趋势，随之产生了各种复杂多样的工具，只有那些一切都从头开始或者根据新的设计流程和方法学而专门成立小组的公司，才能更好地解决改变原有设计方式，用各种复杂工具进行设计所带来的问题。

由于ASIC流程使得许多新公司能够进入IC和系统设计市场，因此，使用这些新工具所要求的资金投入比以前要少，而所要求的专业知识亦更少。

许多大公司都有针对各个层次设计的培训课程，有内部的CAD组来开发设计工具，以及许多研究资源；但小公司也有优势：小公司可以更快地适应新的趋势、新的方法学和设计流程。

小公司没有内部工具程序开发的开销，它必须在资源有限的条件下更富创造性地找到问题的解决方案。小公司必须适应外界开发商所提供的工具，比如Cadence、Mentor、Synopsys和Avant。

这些开发商所开发的工具不是为任何人定制的，它比公司内部开发的CAD工具更能反映市场趋势。而工具开发商并不是完全无约束地进行开发：如果一个公司买了1000套软件包，而另一个公司只买了20套，那么，第一个公司的发言理所当然地更能影响这个工具的新特征。正因为竞争的威胁无处不在，所以只有抢得先机，才能……

让我们简要地列出如今CMOS集成电路设计工程师所面临的主要挑战。我希望把这个前言称为“伞”章，因为从一个工程到下一个工程所出现的问题就像倾盆大雨一样。希望我写的这本书，有助于你们免受淋雨之苦。下面是本书的组织结构介绍。

基础知识

版图设计在芯片开发过程中具体处于何处呢？第1章给出了整个开发过程非技术性的概述，从而使我们能够了解版图设计工程师所起的作用。

IC版图设计者的任务就是创建芯片各个部分的版图掩模，这个版图掩模要符合工程制图要求、网表或者仿真结果以及工艺设计规则的要求。为了理解和遵守工程制图要求，设计者需要了解基本的电学规则以及有关门级版图的概念。这些内容将在第2章中阐述。

第3章讲述了生产工艺和层的定义。在理解了分层是如何被用来组合产生器件和连线后，我们还将学习设计规则，为了确保可以可靠地生产芯片，有一些必须遵守的制造工艺规则。工

艺工程师规定了可生产的最小栅格、多边形、层与层之间的最小间距等。所谓设计规则就是和工程制图、网表等因素一起，从根本上决定芯片结构的一些因素的规则。

版图设计方式

设计者如果不遵守设计要求，芯片就无法工作；如果不遵守设计规则，芯片就可能无法离开原型阶段。优秀的版图设计工程师的高超艺术体现在：在考虑完成时间、最终尺寸、质量等有关工程的常规问题的同时，还能把设计要求和设计规则两者完美地结合起来。

如今，刚才提到的所有芯片里没有一个仅使用了一种设计方式，因此在第5章，我们将讲述设计中的专门技术。我们将讨论在当今IC设计领域中全定制、标准单元、门阵列和其他技术类型，以及各自的优缺点。我们将讲述使用于专用复杂芯片中的各种技术和方法学。所讲述的东西很多，但是有些只是时钟发生器、数据通道或者寄存器组、I/O单元、存储器类型。在第5章的最后还将介绍芯片结尾技术。

高级篇

第6章的议题是与大规模芯片的针对充分连接和电源布线要求相关的，我们将学习针对这些要求的方法学，并讨论布局对布线、版图规划技术和结果、预先规划的信号的影响等。

第7章假设我们已经有了一定的基础并开始解决模拟问题，例如电容器、电迁移和45°版图设计等。

特定工艺要求将在第8章中讲述，既然已经掌握了最复杂的工艺规则，在这一章中，我们将学习宽金属开槽、台阶覆盖、门锁效应和特定的设计规则。

如果设计环境是不确定的，也就是说，还未定义好工艺或者设计仿真覆盖率没有达到100%，这时版图设计人员会面临新的挑战。这就是学习第9章的原因，在这一章中我们将学习将接触孔作为单元、测试pad、备用逻辑和备用线，以及在进行版图设计时记住随时修改。

商业工具

最令人兴奋的可能是第10章。在这一章中，我们分析了所要掌握的各种EDA版图设计工具，以面对各种版图设计所带来的挑战。从生成简单多边形到布局布线，从生成工具及硅编译工具到验证工具，从打印设备及软件到转换格式，我们试图理顺这些名称、概念、方法学以及使用方法。尽管没有评估或者推荐某种特定工具，但仍试图帮助初学者对市场中的商业工具做出恰当选择，并使初学者明白这些工具是怎样适应不同方法学的，或者反之，这些方法学是怎样适应不同工具的。

在世界上，各个公司和工具的名称一直在改变，每年的热门话题各不相同，每年在设计自动化会议上的权威们都会宣布新的重要问题和解决方法，EDA 开发商也因此在不断地提出大量更复杂的设计方法学。写本书目的是帮助读者适应这些方法学，不至于迷失其中。

举个例子，第一代机器的内存是非常小的（CALMA），但随着UNIX的出现则需要更大的内存。还有，伴随验证工具、参数提取工具，以及新名词如深亚微米（DSM）的出现，产生了布局布线工具等。即使这些工具能够解决目前大部分的问题，市场需求（价格）也会造成新的"难以解决的问题"。

本书旨在帮助读者理解这些基础和深层的概念，学会如何分析新方法学以及了解新工具的基本原理。希望本书对所有读者都有所帮助。若有任何批评和建议，笔者将倍感荣幸，来信请寄：

Dan Clein
826 Riddell Avenue North
Ottawa，Ontario
Canada
K2A 2V9
cometic@ieee.org

致　谢

本书与其他书的不同之处在于，在写作的过程中，大家主动花许多时间来解释事情为什么是这样或那样的，它是大家共同交流商讨的结果。我已经尽量列出了所有的“贡献者”，在过去的15年当中，他们帮助我学习理解版图设计中的概念、方法和工具。本书不仅仅是我的书；也是他们的书，因为他们相信教会别人不但可以使自己的生活更愉悦，还可以帮助公司获得更大的成功。这份清单是按年代排序的，与我从他们那里接受信息的数量和重要性未必相关。我对下面这些人致以谢意：

Miriam Gaziel-Zvuloni（英特尔－以色列公司），她看到了我的潜在能力，雇佣我为IC版图设计工程师，尽管那时我几乎不知道希伯来语。就版图知识而言，她是我的第一个老师。

Zehira Sitbon-Dadon，在5年多的时间里，她一直是我的经理，是她促使我进一步学习和研究了更深层的版图概念。她给了我成为版图设计教师、管理项目、负责所有的版图设计工具以及与供应商、工程师、摩托罗拉（以色列）公司CAD小组联系的机会。

Nathan Baron（摩托罗拉－以色列公司），电路设计工程师，他第一个花时间教版图设计工程师在设计电路图时应考虑希望什么、怎样实现、为什么这样。他面对任何新问题最喜欢说的是“首先让我们坐下，慢慢地、慢慢地找到任何问题的解决方法。”

Israel Kashat（摩托罗拉－以色列公司），工程主管，他总是帮忙解决所有进行中的问题，并且总是说：“这是一个多么好的问题啊。我们发现问题是好现象，如果不发现问题、不解决问题，凭什么让别人付给我们工资呢？”

Steve Upham（Cadence－英格兰公司），一个非常热情的应用工程师，他在摩托罗拉（以色列）公司花了5个月的时间去推进新的工具和方法学，他首次向我详细介绍了符号编辑工具和布局布线工具的基本原理。

Carina Ben-Zvi、Nachshon Gal和Eshel Haritan（以前摩托罗拉以色列公司的雇员），他们是CAD工作人员。他们曾和我一起开发各种用于版图设计的内部工具，并多次向我解释软件的局限性、设计理念和基本原理。他们使我对各个行业的软件开发商有了更好的了解。

Jean-Francois Côté（以前MOSAID加拿大公司的雇员），是第一个给我介绍动态随机存储器（DRAM）版图设计技巧的加拿大工程师。他的方法是“我教他人怎样去做我所知道的一些东西，我教得越多，我就要花更多的时间去学习新东西。”我真地相信他说的是对的。

Graham Allan 和 Cormac O'Connell（MOSAID 加拿大公司），是我的存储器设计的指导专家。我今天所知道的与模拟版图有关的版图概念，一些特殊的DRC规则以及DRAM工艺要求多数都来自他们。

Ed Fisher，Mentor Graphics 公司在IC图像多边形编辑工具方面的“权威”，他使我提高了在使用这些工具的能力方面的知识，包括我第一次遇到的器件生成工具

Jim Huntington 是 Cadence 公司验证工具方面的“专家”，他帮助我基于16 Mb的芯片学习、安装并成功地使用了DRACULA工具。

Glenn Thorsthensen，Mentor Graphics公司的应用工程师，他花了许多时间向MOSAID版图设计组解释布局布线工具和压缩工具的使用技巧。

Michael McSherry（Mentor Graphics公司），技术营销人士，他向我介绍了层次化验证的概念和实现。

Steve Shutts（Rockwell公司），第一个比ROSE工具解释得更多的软件开发人员，他向我说明了符号版图工具和版图综合工具是如何使IC设计工程师的工作产生差异的。

Dennis Armstrong（Motorola-Austin公司），版图设计工程师，他后来转向工具的标准检查程序和性能的提高。在过去的10年里，他帮助我理解了各种各样的工具。最初是我在摩托罗拉工作的时候，从那时开始，我们就一直在谈论、交流工具信息。

Dan Asuncion，是位于加利福尼亚Santa Clara的商业与技术学院（IBT）的版图教师，他慷慨地和我共享版图教学经验和他所讲的课程。他是不断鼓励我去写本书的人之一，并且（他）向我保证会将它作为他所教班级的参考书

Mark Swinnen，是Silvar-Lisco公司以前的应用软件工程师，他帮助我理解了关于布局工具、布线工具以及在布局布线环境中要考虑的模拟和数字量的更多内容。

Ron Morgan，是GERED公司的老板，他毫不吝啬地提供给我他们自己培训班的教（课）程，这样就使我能够在已成型（制定）的北美风格的基础上写出加拿大人自己的IC版图教程。

Roger Colbeck 是MOSAID公司半导体分部主管工程的副总，他给了我机会去管理和在加拿大建立第一个经过培训的IC版图设计团队。

Tad Kwasnivski 和 Martin Snelgrove，是位于渥太华的Carleton大学的教授，他们鼓励我去教VLSI学生一些工业上必须了解的知识。这促使我开始了艰苦的写作本书的工作。

Simon Klavey，来自Sagantec 的应用程序工程师，他向我介绍了应用工具方面的所有技巧，并提供了光盘上的演示资料。

Jim Lindauer，来自 Tanner Research，他向我免费提供了一套L-Edit软件，用于本书的写作。特别感谢Tanner Research 提供的一套版图编辑器的演示版本，包括典型的阅览器，这会使读者对IC版图设计有更直观的了解

但是我最感谢的是Gregg Shimokura，本书的技术支持者。我们一起在MOSAID工作的时间超过5年，他总是乐于帮助我和其他人去了解更多的关于VLSI的设计知识。在我们一起工作的5年间，他成为了IC CAD技术部的经理，我们在一起探索能提高设计能力的新方法。写本书的想法由来已久，可真正动笔却是源于他主动向我提供了不少帮助。尽管本书最初是由我执笔的，但却是由Gregg对所有的内容加以排序，他还修正了我的英语，增加了一些与他的工作密切相关的素材：Gregg给这本书增加了工程上的观点。我希望这些观点能帮助学生更多地了解版图的工作结果，从而理解怎样去成为一名更好的工程师。再一次感谢Gregg辛勤的工作，他促成了本书的出版。

目　录

第1章 绪　　论

1.1 专业历史

在过去的20年中，电子工业在规模和复杂度上增长非常迅速。而设计者开始关注芯片设计也不过是仅仅25年前的事情。最初的想法是，设计芯片就是为了减小计算机的体积。其结果是，如今的个人电脑已经取代了曾经有房间那么大的计算机，并且以过去被认为是"不可想像"的速度在运行。集成电路（IC）技术的应用已经深入到我们生活的各个方面。

最初，集成电路版图设计是在一种称为Mylar的特殊纸张上用手工绘制的，这是一项既耗时又费力的工作。市场的需求和技术上的进步，急切要求人们开发出一套软硬件的解决方案来加快芯片的面市时间，尤其是使整个版图设计过程自动化。此外，最终掩模对精确性的要求，也在不断促使版图设计计算机化。

第一个平台是定制的，它能确保图形应用程序快速运行并且有足够的性能。例如，CALMA公司建造了大型计算机，并基于此计算机开发了一套针对印制电路板（PCB）和集成电路（IC）应用的专用软件。

用现在的标准来衡量，该计算机的磁盘空间是相当大的。顶级的主机有220 MB的磁盘空间，并且当时只有0.5 MB的DRAM可以使用。该计算机的标价是100万美元，并且不是每个人都能承受这种设计工作。随着市场和芯片规模的不断增大，有更多公司开始从事芯片设计，而硬件和软件开发者则提出了更快、更小、更经济的解决方案。

在硬件上，最大的变革就是开发出了运行UNIX平台的"工程工作站"。工作站已经发展了多年，它的速度和复杂度是惊人的。因为这些工作站用于进行各种各样的工程设计，所以它们的价格还能承受得起。HP、Sun和IBM是此领域中仅有的几个幸存者，Daisy公司已经从该市场消失了。如今，向更加廉价和通用的平台，如PC机上的Linux和Windows NT平台发展的趋势，为工作站的开发商带来了巨大的压力。

随着硬件平台的不断发展，软件开发也在突飞猛进，它的速度甚至更快。像Mentor Graphics、Cadence、Compass和Daisy这样的大公司，在IC和PCB设计工具方面占有越来越大的市场份额。对于PC平台，如开发出一种称为L-Edit产品的Tanner公司，就是IC设计软件开发市场如何增长的一个例子（在第10章中将有更详细的说明）。

软件开发实际上一直朝着使劳动密集型的工作变得更加自动化的方向发展。例如，不可能用人眼对一个有数百只晶体管模块的设计做互连分析，或者去验证一个256 MB存储器芯片（在第10章中将有更详细的说明）。

下面列出一些关于自动化的重要例子：

版图综合： 版图能够通过“代码”生成，这就替代了手工绘制多边形的传统方法。

版图移植： 通过映射和尖端压缩技术，版图能够交替地从一种设计规则移植到另一种设计规则。

版图验证： 在最终的版图拿去生产之前，需要使用这类工具对版图进行许多的检查，并且这种检查的数量呈不断上升的趋势。例如，检查最小尺寸规则以确保设计是可以进行制造的。

电路综合： 和版图综合很相似，综合过程中，电路原理图能够由专门的“代码”（例如VHDL或者Verilog）自动生成。因为电路完全能够由这些电路综合工具产生，由此产生了对版图自动生成工具更强烈的需求，例如布局布线工具，所以这给版图设计带来了巨大的影响。

布局布线： 完成对差不多上百万个单元的实例布局，以及为了实现连接最小化和电路性能最优化而进行布局优化。

今天，版图设计是在一个不断变化的环境中进行的。软件工具和设计方法，计算平台，工具厂商、客户，正在实现的应用，以及我们所面对的市场压力，所有这一切都在逐年变化着。

所有这些变化已使该行业成为一个令人感兴趣的行业。但不应忘记的是，在制作优质版图后面的基本概念是基于物理特性和电学特性的，这是永远不会改变的。这也是写作本书的一个根本宗旨。

1.2 什么是版图设计

版图设计的定义如下：

版图设计是创建工程制图（网表）的精确的物理描述的过程，而这一物理描述遵守由制造工艺、设计流程以及通过仿真显示为可行的性能要求所带来的一系列约束。

由于在这一定义中蕴涵众多意义，因此，让我们来更加详细地分析这一定义。

过程： 首先，版图设计是一个具有很多步骤的过程，为了得到最优化的结果，这一过程应该遵循一定的逻辑顺序。例如，版图设计的“过程”可以包括：根据适当的分层建立一个数据库或者配置一套工具集；定义每个单元或芯片的版图布局；和/或以适当的顺序，进行验证检查。

创建： “设计”和“创建”通常是同义词，在版图设计中也不例外。在两个不同的工艺上实现同一个电路图，通常所得到的两个版图看起来会很不一样，从而用实例体现出转化实现过

程的创造性。同样，相同的电路图，用在芯片中两个不同的区域，也可能得到不同的结构来适应它所处的具体位置。

精确： 虽然版图设计是一个带有创造性的过程，但是不要忘了最终版图的首要要求必须是，它在一个个晶体管基础上和工程制图是等同的。为了“改善”电路的性能而重新设计晶体管的结构并不是版图设计工程师的任务，除非版图设计工程师同时承担了（或已经承担了）电路设计的任务。

物理描述： 制造CMOS集成电路使用了极端复杂的工艺过程，最后结果是，细小的晶体管和连线是在硅衬底上构建和连接的。版图设计是按照晶体管和连线在硅片上的样子对其进行绘制的艺术。因此，可以认为版图设计是电路的物理描述。

工程制图： 这个名词听起来似乎有一点过时，但是它很确切。晶体管级或者门级电路图在历史上主要是以“图形形式”存在的，并且现在很多公司仍然这么做。现在更复杂的方法学使得一些版图设计工程师可以得到一个称为“网表”的基于文本的文件。尽管如此，为了使人们能够理解网表，通常会附有一个模块级原理图。这些图主要由工程师（或者相应的人）提供，但随着产业的发展，它也会相应地改变。

遵守： 遵守，我们指的是“满足……的要求”，而不一定是“尽可能最小或者最好的设计”。在设计过程中需要做很多的折中，如可靠性、可制造性、灵活性以及（可能是最重要的）上市时间。当然，有很多最低要求必须被满足，但在当今的市场中，以牺牲项目进度的代价来优化设计是不实际的。

制造工艺所带来的约束： 这些约束包括如金属线的最小线宽之类的版图设计规则，但是也有许多关于可制造性或者可靠性方面的其他准则，而这些准则能提高版图的总体质量。比如，以金属线为例，更宽的金属线宽可以使设计更具有可制造性。因此，在间距允许的条件下就应该使用更宽的线宽。

设计流程所带来的约束： 这些约束包含建立一系列准则，建立这些准则的目的是为了使在设计流程中用到的所有其他工具能有效地用于整个版图。例如，一些布线工具倾向于以标准间距连接单元，而其他的布线工具则不关心。另外一个例子是为版图加入文字标识的方法，以便于这些文字可以在后续部分中用做标识。

通过仿真显示为可行的性能要求所带来的约束： 工程师如果不知道版图中电路实现的细节，要完成一个电路设计，那么就需要做一些假定。例如，在版图实现之前，电路设计工程师无法知道模块的准确面积，因此这就需要基于已有的信息做一些有根据的估计。总的面积可能是很重要的，知道了总的面积，也就知道了模块内的最大线长。通常，这些假定是无法避免的，而诀窍就是尝试将这些假定相互传递，并由此来对版图进行相应的约束。在本例中，电路设计者所使用的总面积的估计值，也应被版图设计工程师作为目标面积来使用。对这一估计或低或高的偏差应反馈给电路设计工程师，以便重新进行仿真。

总之，版图设计包括许多不同的领域，它需要许多不同的技能，并且需要做出许多影响最终实现的质量的折中和决定。优秀的版图设计需要对以上所有这些议题都有充分的理解。在本书中，我们希望以不同的程度来涵盖所有这些内容。

1.3 IC 设计流程

版图设计在整个设计流程中处在何种位置呢？正如1.2节中所说明的那样，一旦工程制图完成了，版图设计也就开始了。我们可以通过IC设计周期中的前后关系，了解一下版图设计，以及它在“流程”中处于什么位置。

在开发过程中，有多种基于特定设计的设计流程。这里考虑一个通用概念上的流程，通过此流程，则所有的产品方案可以实现，产品最终能进入市场（如图1.1所示）。

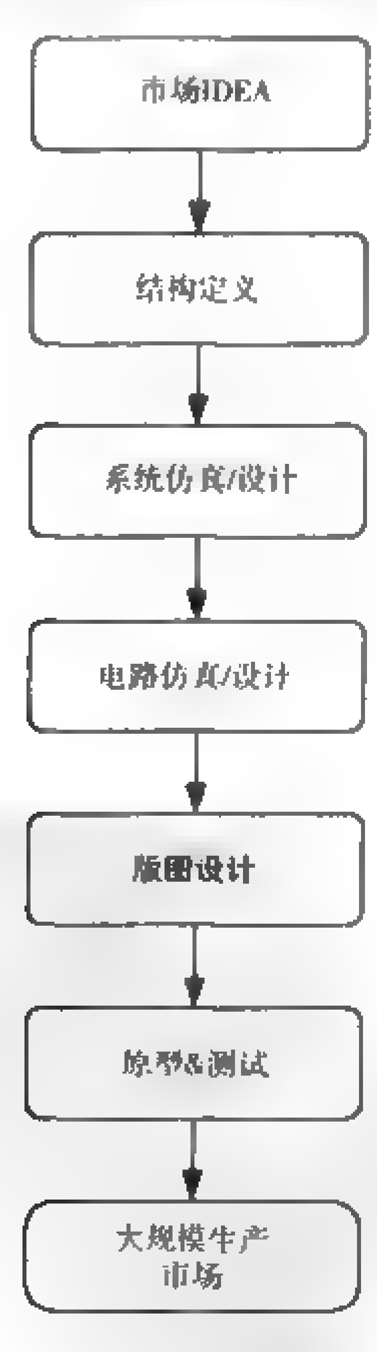


图1.1 IC 设计流程

1. 首先，市场部通常会详细说明需要开发的产品。
2. 下一步是规定设计的结构或者行为。电路设计工程师规定芯片的结构来满足市场和/或IDEA功能需求。
3. 系统仿真由一组工程师完成。这组工程师会对将要集成在最终芯片中的各个单独模块进行定义和验证。这一步骤验证了在步骤2中规定的结构是合理的，并清楚地定义了需要进一步实现的可处理的模块。
4. 电路设计组完成所有的数字和模拟仿真，来验证电路的方案和门的连通性，以及门的尺寸（为了满足时序规范）。这些组需要和版图设计组进行交互，版图设计组会使电路适合芯片的版图布局。
5. 版图设计由版图设计工程师完成。他们的工作包括放置多边形，对于所有的模块，利用电路组生成的电路图来实现晶体管、基底连接、连线（使用1至6层金属）等。拿去大规模生产的最终设计是整个芯片的版图。
6. 在第一块晶圆制造出来后，测试工程师组就要开始尝试测试芯片。首先，他们将检查工艺参数是否在可以接受的允许误差范围内。下一步是使用工程测试仪来测试芯片，以便于找出所有的违规，并尝试在现场解决这些问题。
7. 在改正所有的错误（工艺上的和/或逻辑上的）后，芯片就将开始批量生产并流入市场。

应当记住，这只是一个概念上的流程。在实际中，在通过不同的设计阶段时，会有很多的反复和设计上的迭代。很多不同的因素会引起设计上的改变，包括很多来自于版图上的限制或者约束。在问题发生之前就能预料到这些问题，是理解如何区分极优秀的设计工程师和优良的设计工程师的基本原则所在。

从哪里开始呢？从版图设计工程师的角度来看，这些工作是从创建了原理图或者网表之后开始的。详见第2章。

第2章　电路图基础

如果给你一个电路图，或者你已设计了一个电路图，并准备去进行版图设计。那么接下来该怎么办？在这一章中，我们将学习电路图的基本构建模块，以及实现版图设计所需要准备的基础知识。让我们首先介绍所有CMOS电路的基本构建模块——晶体管；然后说明什么是典型电路示意图，并以此为基础介绍更深的知识。

2.1　MOS晶体管：基本电路结构

为了有效地进行版图设计，晶体管是我们所需要了解的最小的构建模块或器件。让我们首先了解晶体管的功能特性，并基本理解晶体管的工作原理，这样就可以最大限度地提高设计的性能。

CMOS指互补金属氧化物半导体。这个命名是比较恰当的，因为在CMOS工艺中有两种具有不同特性的晶体管，即PMOS晶体管和NMOS晶体管，它们组合在一起，两者互为补充，具体情况见本节中的介绍。图2.1给出了PMOS晶体管和NMOS晶体管的典型电路图表示。请注意，在这两幅图形中，所画的源极和漏极节点是相反的



图2.1　PMOS晶体管和NMOS晶体管

在大多数情况下，对于PMOS晶体管来说，“基底”总是连接到逻辑“1”电平；而对NMOS晶体管来说，“基底”总是连接到逻辑“0”电平。因此，大多数电路图都不会给出基底连接；但是，它是隐含在其中的。当然，也并不总是如此　在本书后续的电路图中都将暂时省略“基底”连接。

PMOS晶体管和NMOS晶体管的栅极导通或者说晶体管“导通”的条件是不一样的。当PMOS晶体管的栅极处于逻辑“0”电平时，晶体管导通。相反，当NMOS晶体管的栅极处于逻辑“1”电平时，晶体管导通。记住这一点的技巧是，PMOS晶体管的栅极上的小圆圈看起来像是一个“0”，而NMOS晶体管的栅极看起来就像是一个“1”（见图2.2）。



图 2.2 PMOS 和 NMOS 晶体管栅极导通

两种晶体管的工作方式，很像是“开关”或是水管上的阀门。如阀门那样，“栅极”控制着开关的导通或者关断。正向电流被定义为：当栅极导通时，“泄出”的“水”或者说电荷从晶体管的漏极一侧流向水源或者说源极一侧的操作。如果栅极关断，那么电流（或水流）就不会流动。

当晶体管导通的时候，可以用一种更简单的方式将其想像成一个电阻器（见图 2.3）。



图 2.3 PMOS 晶体管电阻模型和 NMOS 晶体管电阻模型

流过晶体管的电流总量受到晶体管等效电阻的限制。我们将在后面谈到，晶体管的尺寸会直接影响它的等效电阻。基于这一认识，我们将用这个简化的电阻模型来分析晶体管的工作方式。

现在让我们来考虑晶体管的源极连接到一个恒定的逻辑电平的情形。通常，在电路图中，逻辑“1”电平由设计中最高的电源电压来表示，这一高电源电压通常会被标注为 VDD、VCC 或者也可能是 VPP。反之，在电路图中，逻辑“0”电平由芯片的地电平来表示。典型的命名为 VSS、GND 或者 GROUND。在这种情况下，如果晶体管的栅极导通，那么漏极就会自然地驱动到和源极相同的电平。

由于 PMOS、NMOS 器件的相关物理特性和局限性（不在这里讨论），PMOS 晶体管几乎总是用于建立逻辑“1”电平，而 NMOS 管则用于建立逻辑“0”电平（见图 2.4），尽管也会有一些例外的情形。因为将 PMOS 晶体管和 NMOS 晶体管组合在一起，能够简单而又可靠地生成两种逻辑电平，所以它们是互为补偿的，这就是之所以将它们组合在一起并称为“互补”的原因所在。并且正因为如此，布尔逻辑可以很容易地由 PMOS 晶体管和 NMOS 晶体管实现，这也是 CMOS 电路现在如此流行的主要原因之一。



图2.4　PMOS晶体管生成“1”和NMOS晶体管生成“0”

不要完全忘记了前面提到的基底连接。记住，基底通常连接到相应的逻辑电平上，图2.5给出了和相应电源电平隐含的连接。

晶体管的尺寸也应该在电路图中标识出来（见图2.6）。每一个PMOS晶体管和NMOS晶体管都有长度和宽度。在后面的章节中，我们将详细解释这些尺寸参数，而现在我们认为已经给出了这些参数。通常，两种晶体管的长度不会被标识出来，而是有一个默认值。这一默认值通常是由加工工艺所限制的最小允许值，并且这一数字也会被引用来指定工艺。例如，0.25 μm工艺特指默认的栅极长度是0.25 μm。由于它是一个冗余信息，因此不在电路图中给出。



图2.5　显示隐含基底连接的MOS晶体管



图2.6　标注器件尺寸的MOS晶体管符号

在图2.6中，PMOS晶体管的宽度是5 μm，而NMOS晶体管的宽度是10 μm。宽度的值通常放在前面。在图中，PMOS晶体管的长度是0.5 μm，而NMOS晶体管的长度并没有给出，因而就可以认为其长度是工艺的默认值——0.25 μm。

只要考虑晶体管的版图设计，事情将变得更加显而易见，即，当晶体管的宽度增加或者长度减小时，晶体管的电阻将会减小，而晶体管的电流驱动将会增加。对于本章，请视此为已给的内容。

2.2　逻辑门

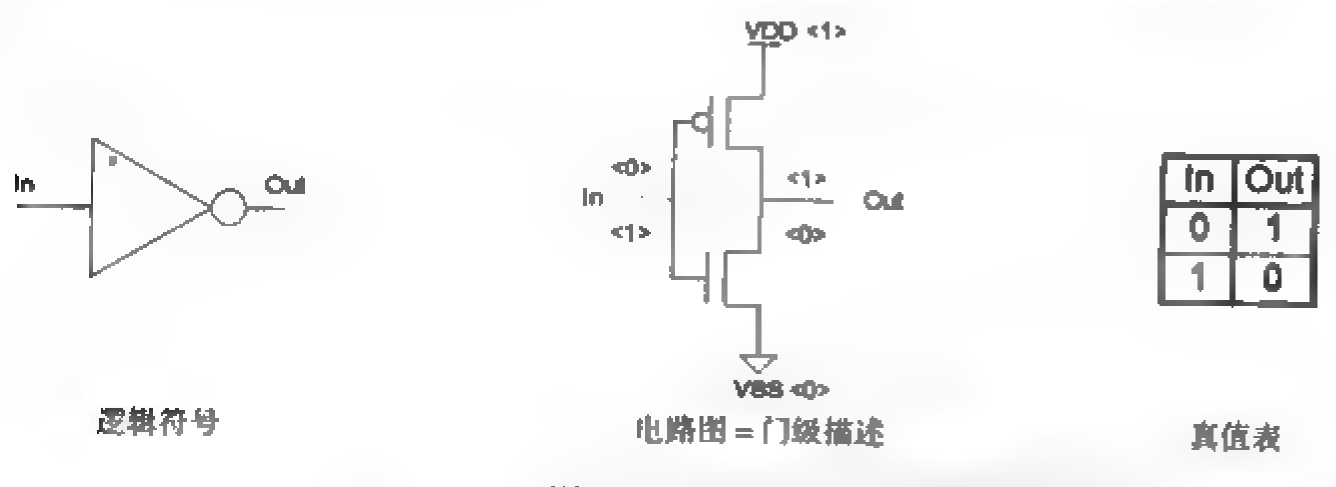
如今大多数的电路图上并不是布满了晶体管。造成这种情形的原因有很多，但是其中一个主要的原因是由于所承担的设计复杂而使这样做不切实际。因此，晶体管都被组合成所谓的逻辑门或者“门”。逻辑门（gate）可能会和晶体管的栅极（gate）混淆，但是我们希望使用该术语的上下文足以让人明白其具体含义。

逻辑门可以用来直接或者组合形成布尔逻辑函数。理论上，几乎任何布尔逻辑函数都可以由单个逻辑门来实现，但是在实际中并不这么做。我们希望读者在读完本书之后会完全明白其中的原因。

一般来说，大多数CMOS的逻辑函数都可以用反相器、两输入到四输入与非门（NAND）、两输入到四输入或非门（NOR）以及传输门来实现。让我们从最简单的逻辑门——反相器开始来学习这些逻辑门。

2.2.1　反相器

顾名思义，反相器是最简单的逻辑门。它的功能就是将在输入节点上接收到的信号反相为相反的极性并从输出节点输出（见图2.7）。



In	Out
0	1
1	0

图2.7　反相器

晶体管知识告诉我们，当接收到的输入信号为“0”时，PMOS管“导通”，这意味着输出端是“1”。在这种情况下，NMOS晶体管是关断的，它不影响输出电平。反之，同理可得，当输入为“1”时，输出为“0”（见图2.8）。

CMOS逻辑由于自身的特性总是为反相的。同样要注意的是，NMOS晶体管和PMOS晶体管永远不可能同时导通。这就说明了CMOS电路为什么是一种低功耗形式的电路设计的原因。当栅极改变状态时，在VDD和VSS之间并没有直流（DC）电流通路；但是，如果存在着这样的一条通路，就会有直流功耗。

要指定反相器的尺寸，需要两种器件尺寸（见图2.9）：

- 标识符“P”和“N”指定了器件的类型。此外，通常会先给出宽度。
- 在图中，PMOS 晶体管的宽度是 2 μm，而 NMOS 晶体管的宽度是 1 μm。
- PMOS 晶体管的长度是 0.5 μm，而由于 NMOS 晶体管的长度并没有给出，因此可以认为其长度为工艺的默认值。

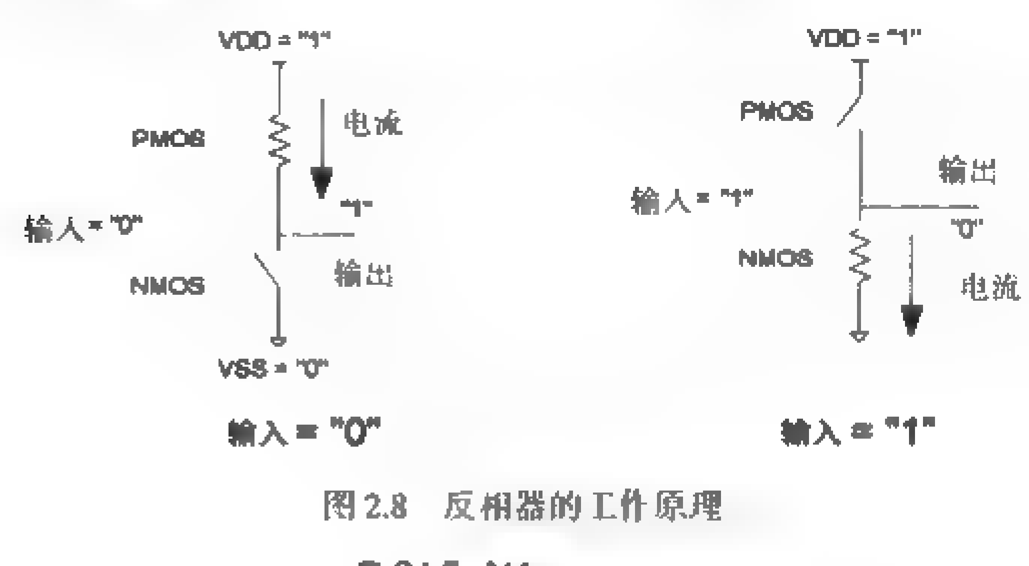


图 2.8 反相器的工作原理

P 2/.5, N1

In

Out

图 2.9 反相器的尺寸

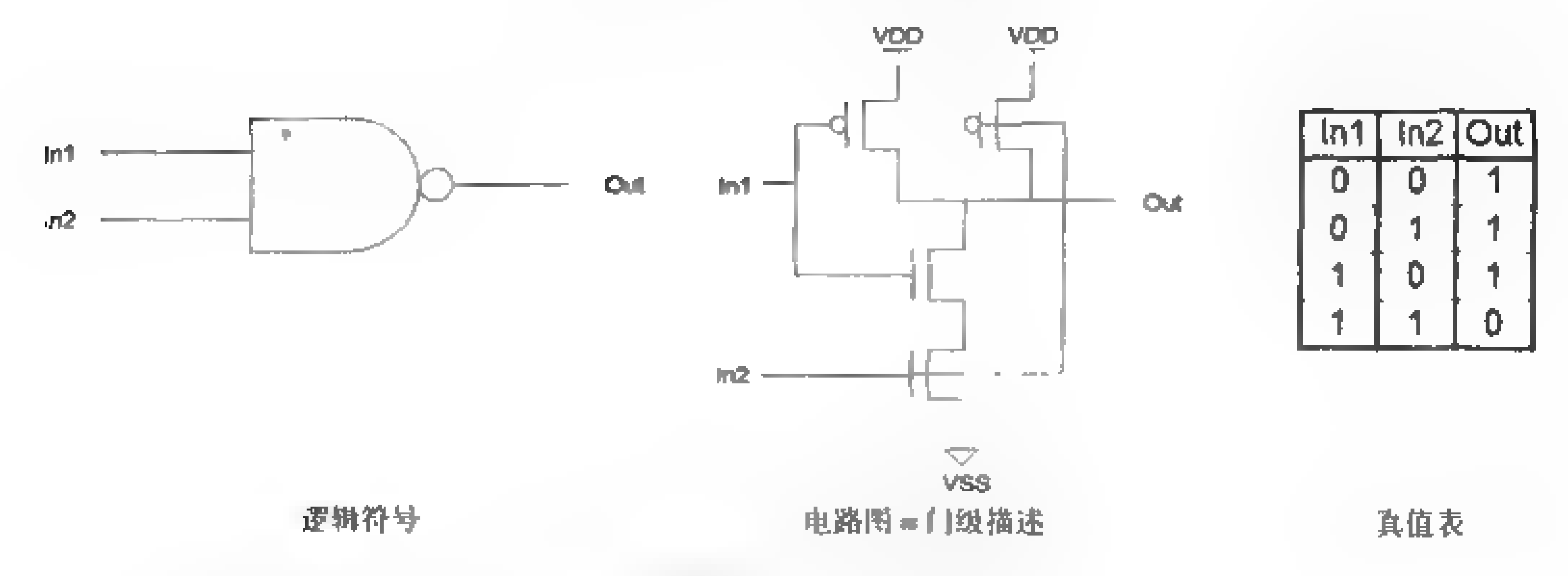
下一节将谈到与非门（NAND）和或非门（NOR）。与非门（NAND）是与门（AND）的反相，而或非门（NOR）是或门（OR）的反相。它们都是单级门，而这正是它们成为 CMOS 逻辑的基本构建模块的原因之一。

2.2.2 两输入与非门（NAND）

当需要在两个不同的信号之间做出逻辑判定时，就会使用到与非门和或非门。若遵循在图2.10给出的真值表中的每个输入条件下单个晶体管的操作，则用图中给出的晶体管结构就可以得到期望的输出。

“与非”的功能（输出信号 OUT = “0”）是在输入信号 IN1 和 IN2 均为“1”时得到的。当将两个 NMOS 晶体管串联连接时，就能满足输入同时为“1”的要求。与此同时，PMOS 晶体管则以一种互补的形式并联地连接。

这种结构不仅能通过用逻辑门来产生正确的逻辑功能，还可以确保到 VDD 的 PMOS 通路和到 VSS 的 NMOS 通路这两条通路永远不会同时导通，从而降低了静态直流功耗。



In1	In2	Out
0	0	1
0	1	1
1	0	1
1	1	0

图2.10 两输入与非门（NAND）

三输入或更多输入的与非门（NAND）可以很容易地通过扩展NMOS晶体管的串联连接和PMOS管的并联连接来实现。

若要指定与非门晶体管的尺寸，则需要四个器件的尺寸。但在多数情况下，所有的PMOS晶体管尺寸相同，类似，所有的NMOS晶体管的尺寸也相同；因此，通常只需要两个参数值即可（见图2.11）。对于或非门（NOR）也是如此，这表明或非门（NOR）的尺寸也用类似的方式处理。

- 标识符“P”和“N”指定了器件的类型。此外，通常会先给出宽度。
- 在图2.11中，PMOS晶体管的宽度是15 μm，NMOS晶体管的宽度是5 μm。
- PMOS晶体管和NMOS晶体管的长度可以认为是工艺的默认值。

当两个独立的PMOS晶体管要求有不同的尺寸时，通常在标识符“P”后面加上一个数字，比如“P1, P2”，然后给出附加的参数值。

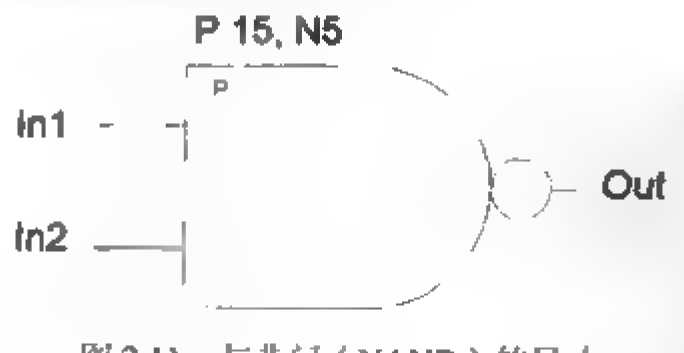
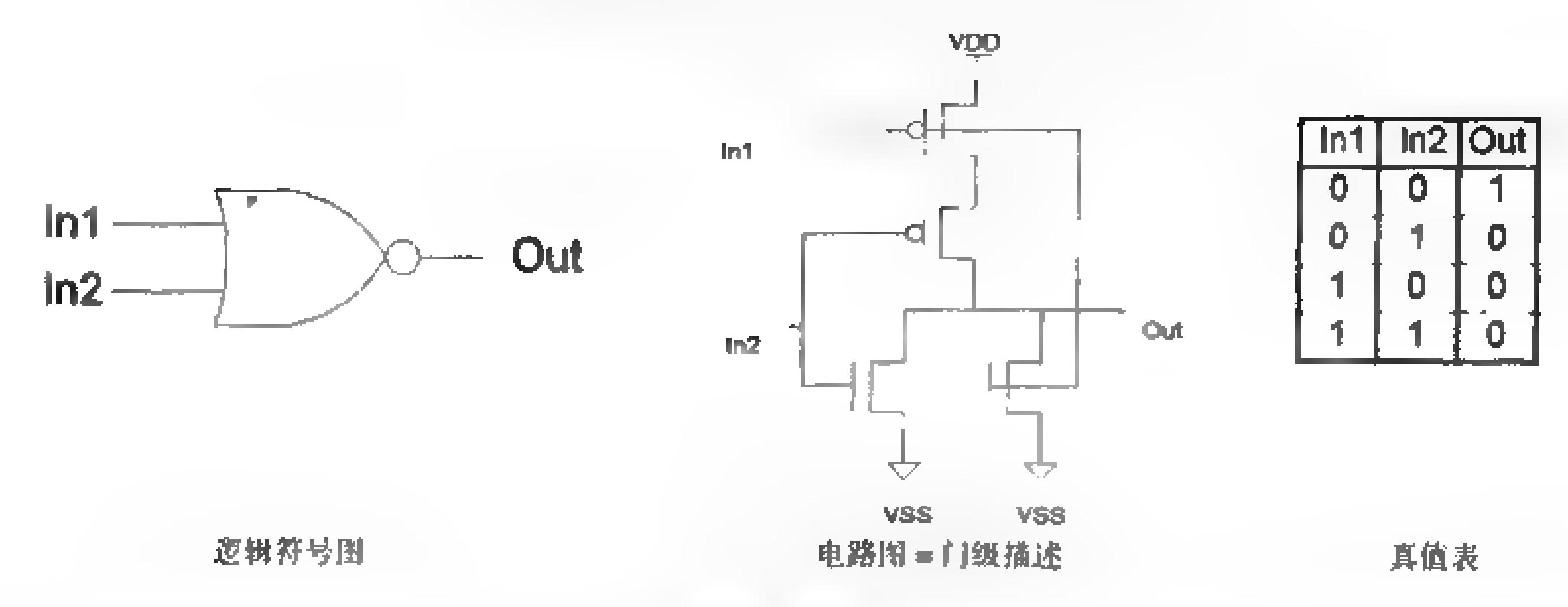


图2.11 与非门（NAND）的尺寸

2.2.3 两输入或非门（NOR）

或非门（NOR）是与非门（NAND）的镜像或互补性结构。在或非门（NOR）中，NMOS晶体管和PMOS晶体管之间的串/并联连接方式和与非门中的连接方式相反——PMOS晶体管是串联方式，而NMOS晶体管是并联方式（见图2.12）。



In1	In2	Out
0	0	1
0	1	0
1	0	0
1	1	0

图2.12　两输入或非门（NOR）

同样，对于所有的输入情况，直流功耗的可能性都被消除了，并且，通过分别增加PMOS晶体管和NMOS晶体管的串并联连接，可以很容易地得到三输入或多输入变量的或非门（NOR）。

晶体管尺寸数值的表示方式和与非门（NAND）相类似，因此对于一些特定的约定，在此不再赘述。

2.2.4　复杂逻辑门

如前所述，几乎任何布尔逻辑函数都可以通过一个单级的CMOS逻辑门来实现。所谓复杂逻辑门，是指那些具有“复杂的”逻辑功能的逻辑门，通常是将与（AND）、或（OR）、与非（NAND）以及或非（NOR）组合起来，而所有的这些组合在同一逻辑级中实现。

因为复杂逻辑门在单级中实现，所以在几乎所有的情形中，都可以获得功耗、面积以及速度上的优化。

图2.13是由多个门实现一个复杂逻辑函数的例子。

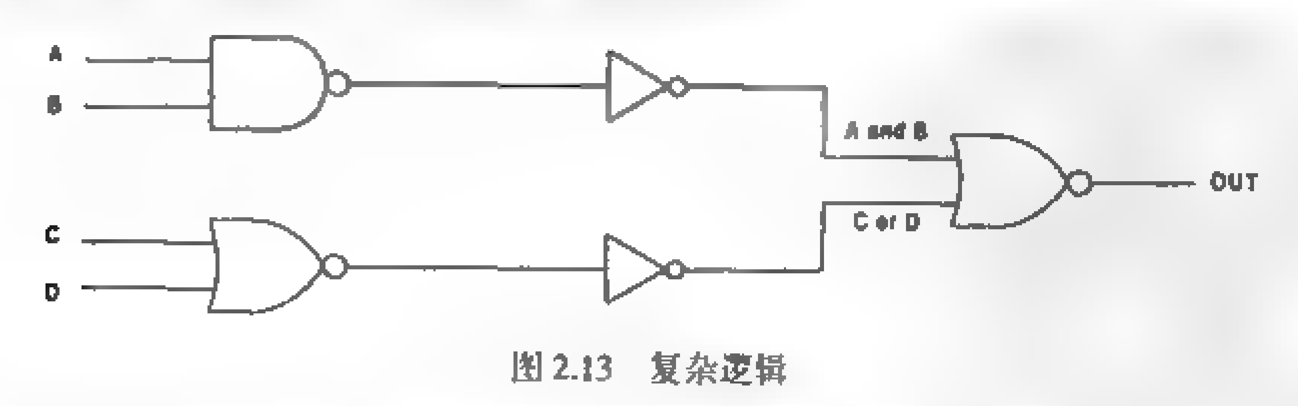


图2.13　复杂逻辑

如果我们对这一逻辑做一次简单的晶体管统计，就会发现在全部的三级逻辑中共含有16个晶体管。通常，不会用这种方法来设计工程电路图，而是用图2.14所示的用一个符号表示单级逻辑的方法。

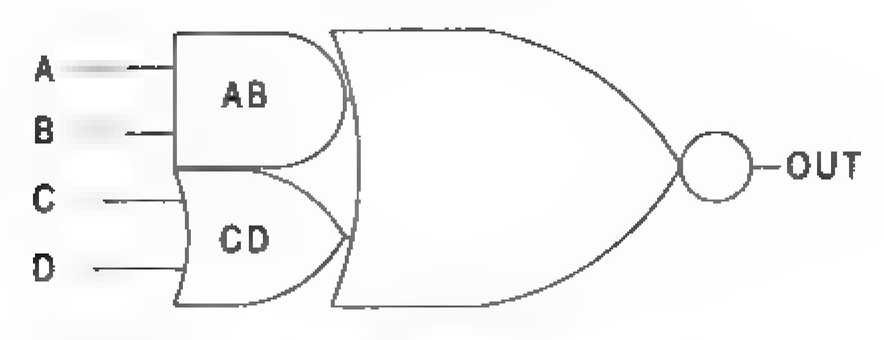


图2.14 复杂逻辑门示例

可以看到，通过将反相器和它们各自的驱动门组合在一起，与非门-反相器组合成为一个与门（AND），而或非门-反相器则组合成为一个或门（OR）。或非门（NOR）的输出仍然保持不变。

那么这个逻辑门的晶体管级描述是什么样的呢？我们需要用这一描述来进行版图设计。

这种类型的复杂逻辑门使用方便、构建快捷，但在绘制时或多或少会有些麻烦，为了确定晶体管级描述，我们从输出端的逻辑门开始分析，然后往后倒推（即从右到左）。

首先考虑两输入或非门（NOR）的输出端．想法是这样的，用一个与非（NAND）逻辑来代替图2.14中的与门（AND），同时用一个或非（NOR）逻辑来代替图中的或门（OR），然后将这两个逻辑组合成或非门（NOR）的输出，从而创建最终的复杂逻辑门。

为什么我们要用与非（NAND）输入而不用与（AND）输入呢？同样，为什么用或非（NOR）输入而不是或（OR）输入呢？

答案是这样的，或非门（NOR）输出提供一个额外的逻辑反相级，我们可以在实现最终的逻辑门时利用这一点。因为在或非门（NOR）输出中有一个内在的反相逻辑，所以我们不需要实现输入与（AND）或者或（OR）逻辑功能；而与非（NAND）和或非（NOR）逻辑功能恰恰是我们所需要的。读者自己来完成这一工作并且验证一下这一原理会受益匪浅。

在将晶体管进行如后面所述的合并之前，我们要做这样的准备步骤，即确定将输入端的逻辑门并入输出端逻辑门。通过对输入端的逻辑进行反相可以很容易地完成这一步骤。在这里，我们将与（AND）逻辑反相为与非（NAND）逻辑，而或（OR）逻辑则反相为或非（NOR）逻辑。

1. 我们用与非（NAND）逻辑输入中的并联PMOS晶体管替换PMOS晶体管AB，用相应的串联NMOS晶体管替换NMOS晶体管AB。
2. 然后，我们用同样的方法对器件CD进行替换，用或非（NOR）逻辑输入中的串联PMOS晶体管替换PMOS晶体管CD，而用相应的并联NMOS晶体管替换NMOS晶体管CD。这样就得到了我们想要的结果（见图2.15）。

如果对最终结构的真值表进行校验，可以发现这个8晶体管的逻辑门在逻辑上和前面的16晶体管、三级的逻辑门是等同的。

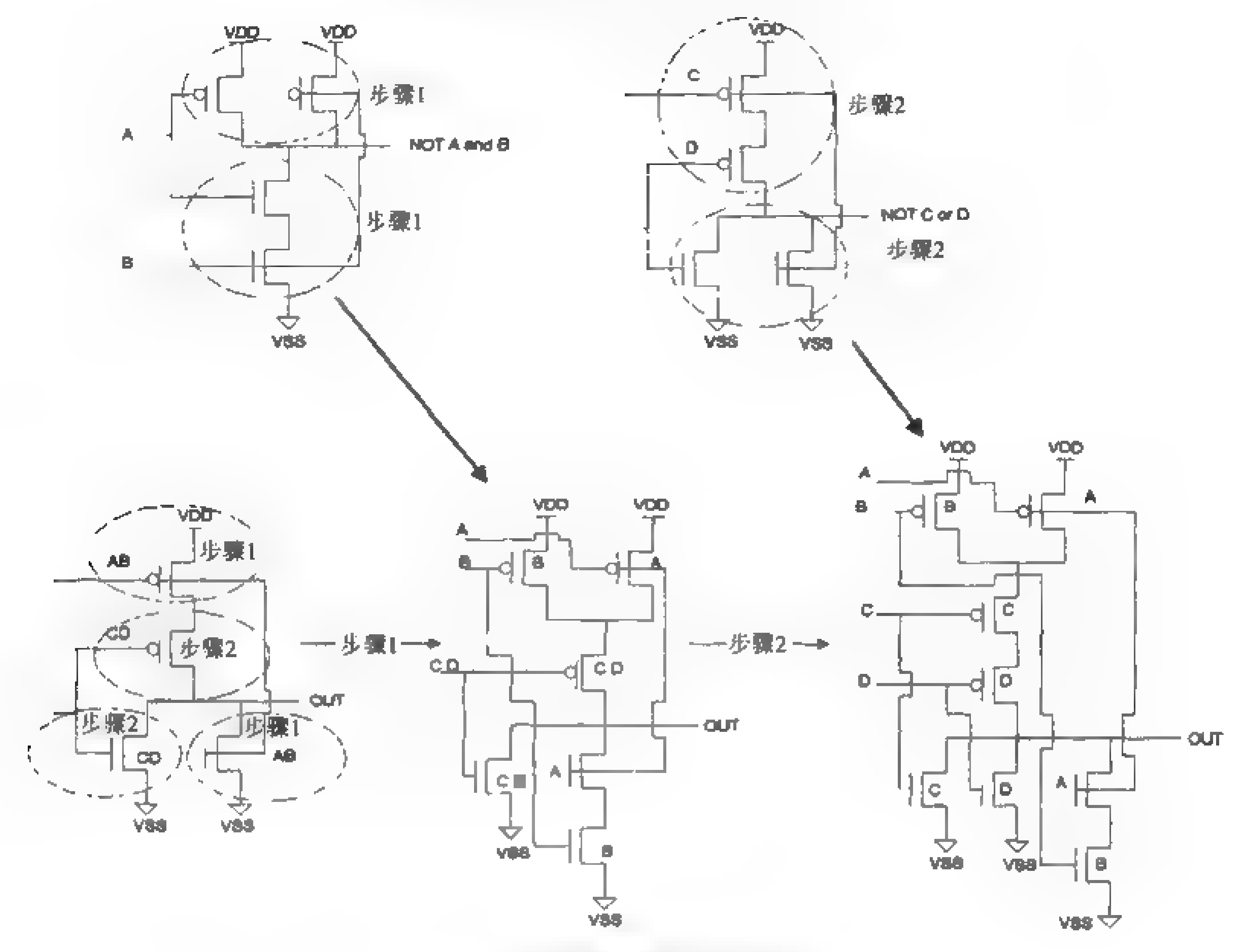


图2.15 复杂逻辑门的解决方案

使用这种方法可以推广和理解复杂逻辑门的简化方式!

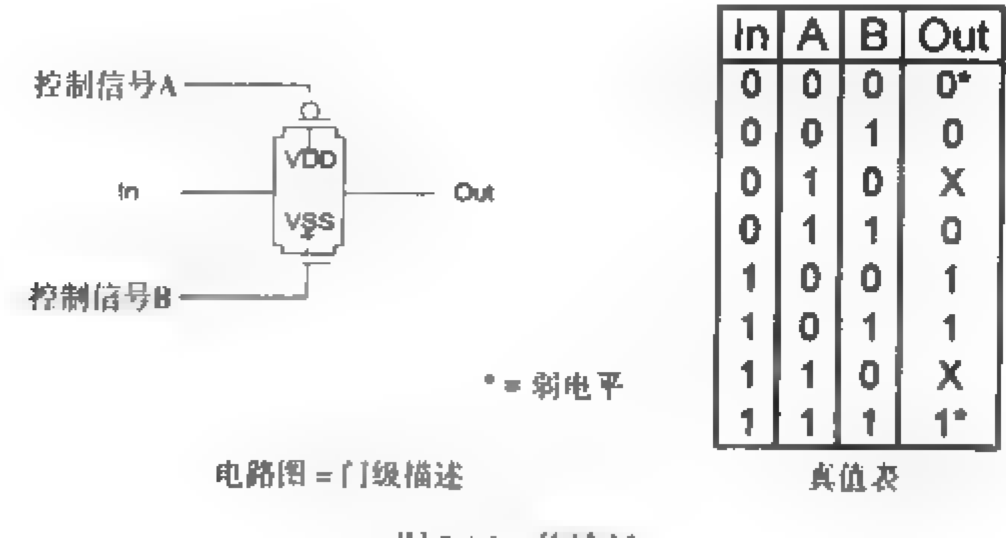
由于典型的复杂逻辑门有更多的晶体管，因此，在电路图中，我们可以将单个晶体管尺寸标识出来，也可以不将单个晶体管的尺寸标识出来。在大多数情况下，每个晶体管都会有不同的尺寸，因此在符号图上晶体管的尺寸通常被省略。必须通过查看晶体管级电路图才能确定晶体管的尺寸信息。即使标识了器件尺寸，在进行版图设计之前，将这些尺寸映射到晶体管结构时，仍然需要进行手工检查。

2.3 传输门

现在让我们来研究另一种晶体管结构，它也会在电路图中出现。

对于反相器来说，两个晶体管的源极都连接到电源端。对于组合逻辑门来说，串联连接的晶体管形成链的一部分，这条链最终连接到一个电源端。因此，应以类似于处理简单反相器的方式来处理这些晶体管。

传输门是一种非常常见的逻辑门类型，它的源极和漏极均作为信号线来使用。在这种情况下，输出信号通常基于控制信号A和B的状态来跟随输入信号。需注意的是，这种结构只考虑输入信号的非反相传播，当控制信号禁止PMOS晶体管和NMOS晶体管时，输入信号则被封锁，这些正是这种逻辑门的强大特性。传输门的使用非常频繁，因而我们需要仔细地对它进行设计（见图2.16）。



In	A	B	Out
0	0	0	0*
0	0	1	0
0	1	0	X
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	X
1	1	1	1*

图2.16 传输门

记得前面说过，通常PMOS晶体管连接成用来产生逻辑“1”电平，而NMOS晶体管用来产生逻辑“0”电平，几乎从未有相反的情况。图2.16中给出的传输门的真值表说明了为什么会是这样的原因之一。PMOS晶体管能够传输“0”电平，但是它们有些不情愿，从而削弱了“0”电平。对于NMOS晶体管传输“1”电平的情况也是如此。这就是在真值表中“弱电平”的意思。除非做特殊考虑，在加强的逻辑设计中，通常要避免这些弱电平存在的情况。通常情况是，控制信号控制传输门不是完全“导通”就是完全“关断”（对于两个晶体管均是如此），而不存在一种中间状态。

2.4 理解电路图连接关系

在实现任何电路图的版图设计时，到最终设计之前，存在着比明确给出的连接关系更多的连接情况。在电路图上显示的连接只不过是绘制的从A点到B点的一条简单连线，或者是两个晶体管以串联或并联方式连接的一种简单连接关系。而在实际中，一条连线代表了一条需要在物理上实现和优化的信号线。下面来看一个例子（见图2.17）。

图中逻辑门和晶体管看起来应该非常熟悉，并且对各种逻辑门的不同的晶体管描述已经给出了说明。现在的难点在于理解器件的连接关系。我们已经知道，连接到每个晶体管的基底连接是必需的，但是在电路图中并没有表示出来。表2.1对各种不同类型的符号进行了概述。

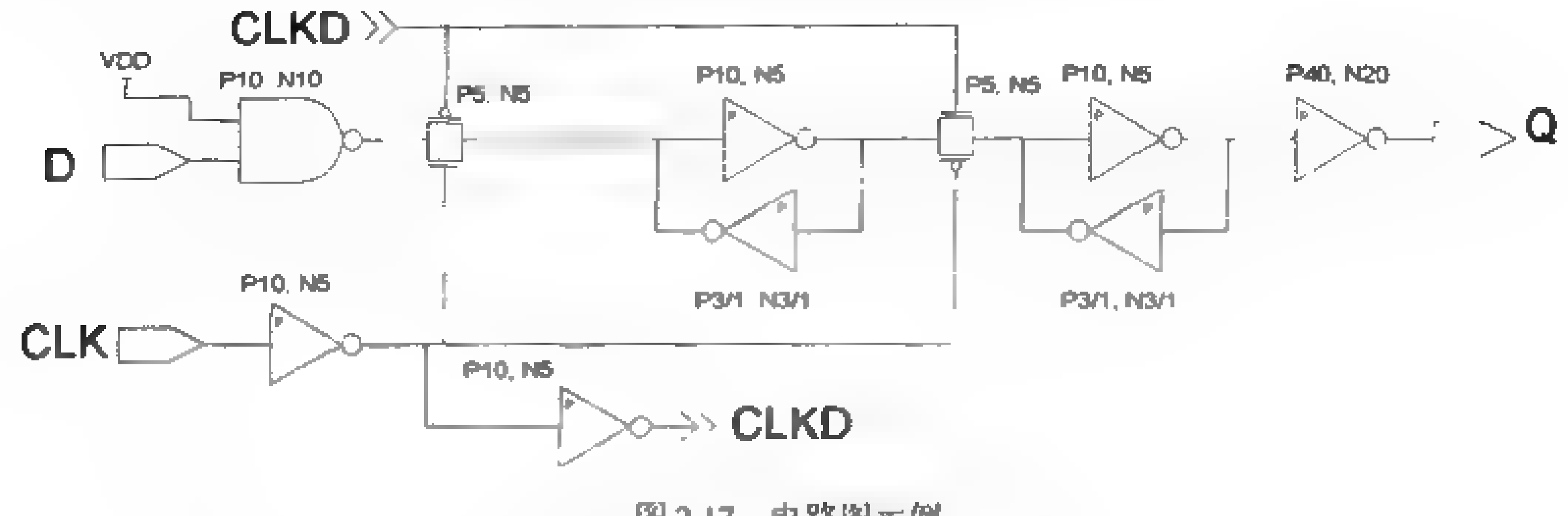


图2.17　电路图示例

表2.1　电路图连接

电路图符号	描述
——	简单线连接。这些信号是局部信号，并且需要考虑的是，在一张电路图内对它们进行布线和实现
>>	页内连接端子。此符号完成一个虚拟的连接。连线名或者节点名用来标识在电路图中此线网的布线位置。在本例中两个节点标识为CLKD，它们在电气上是连接的，但是在图上无法看到该连接。通常这样做的目的是避免电路图上的连线太过繁杂，从而将原理图弄乱
▭>	端口或者引脚连接端子。这个符号表示线网在电路图中是输入还是输出，它是电路图和外部“接口”的一部分。出于性能和可靠性的原因，这些信号可能会附有一些特殊考虑，因此，查找这些情况是否存在是非常重要的
VDD	全局连线端子。我们把它看做是晶体管的基底连接。全局连接端子标注出了电路图模块内部以及外部所要求的电气节点。图中所示的“VDD”线网是全局性信号，任何地方都可使用。另外，把所有隐含的内部连线都画出来是不现实的

2.5　回顾基本电学定律

集成电路版图设计其实是一门用多边形和各种形状来实现电路的艺术，而这些多边形和形状描述了晶体管和连接关系并形成最终的设计，必须牢记的一个重要概念是，物理版图的特性在很大程度上决定了最终设计的电学特性。

本节的目的是回顾几个必须掌握的基本的电学定律和原理，从而使我们打下一个良好的基础，并基于此逐步获得高效而有效的版图设计方法。

2.5.1　欧姆定律

这是最基本和最重要的定律：

$$V = I \times R$$

电压＝电流×电阻

我们知道，在MOS晶体管“导通”或栅极“导通”时，其工作可以等效为“电阻”。由栅极导通所感生的电流会引起晶体管两端的电压变化。这恰恰论证了欧姆定律的运用！对于给定的晶体管的电阻值以及正向电流值，产生的电压变化可以用欧姆定律来解释（见图2.18）。

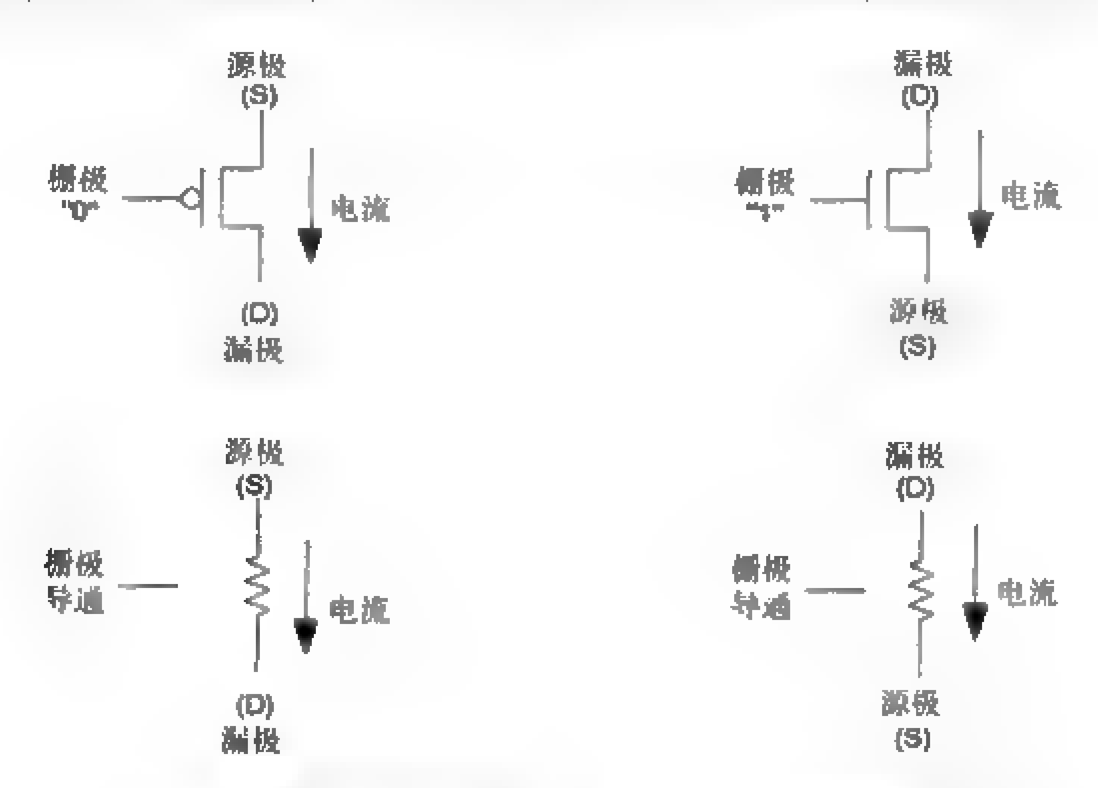


图2.18　PMOS晶体管模型（左）和NMOS晶体管模型（右）

类似地，当栅极“关断”时，电流为“0”。根据欧姆定律，电压的变化也是零，可以这样理解，当栅极“关断”时，晶体管的作用就像电路开路。

实际上，晶体管的电阻值是动态值，流经晶体管的电流总量也是如此。因此，这是一个非常简单化的模型，但它有效地解释了欧姆定律是如何起作用的，并且让我们了解了在晶体管如何工作的后面所蕴涵的基本原理。

欧姆定律是一个应该记住且非常有用的定律，它是电路的基础，对于版图设计同样如此。

2.5.2　基尔霍夫电流定律

基尔霍夫电流定律是另一个基本定律，它可以用来解释后续章节中的某些概念。基尔霍夫电流定律所表述的是，流入任一电学节点的电流的代数和为零。在图2.19中，我们根据惯例将流入节点的电流视为正向电流，而将流出节点的电流视为负向电流，因此它们的总和应该等于零：

$$I_1 + I_2 + I_3 + \cdots + I_N = 0$$

基尔霍夫电流定律的另一种表述方法是，流入节点的电流总和必等于流出节点的电流总和（见图2.19）。

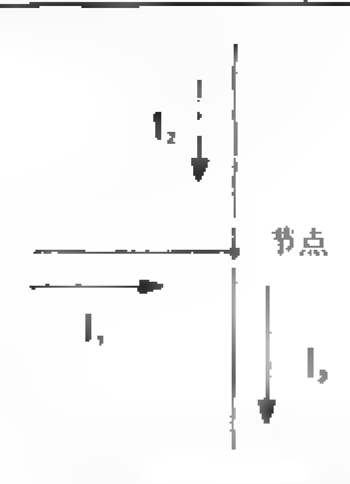


图2.19 节点电流和基尔霍夫电流定律

2.5.3 电阻

前面提到了电阻的概念，但是没有给出更详细的解释。在晶体管为“导通”状态时，我们用电阻来对其进行建模。

简单地说，电阻可以认为是导体导电的阻力（或能力）。用水来做类比，直径较大的管道比直径较小的管道具有更小的阻力，因为它能够通过更大的流水量。此时，直径大的管道的截面面积比较大。这里假定了两根管道具有相同的长度。而当管道或者导体的长度增加时，其阻值也会随之增加。

在集成电路（IC）设计中约定，导电层的电阻值计算是用每“平方面积”的阻值来表示的。一个“平方面积”定义为当导体长度等于宽度时的面积。

计算导体阻值的公式为

$$R = \rho \times l/w$$

其中，“ρ”是导体层的电阻率，单位是Ω/□；l是长度，而w是导体的宽度。

2.5.4 电容

简而言之，电容可以看做是在指定节点和参考节点之间每单位电压一个物体或导体所能保持的电荷总量。用水来做类比，电容器可以看做是拦截湖，根据用电设备所需的电能而将其装满水或排空水。

导体电容值的大小取决于该导体的面积以及它和参考节点之间的距离。再用水做一个类比，假设有一个湖，充满它需要用多少水（想像对电容器充电需要多少电荷）？这当然要取决于湖面的面积以及湖的深度。

这一概念中难以理解的部分在于，参考节点即湖底，和湖面之间的距离决定了湖的深度。参考节点离导体越远，湖水就越浅。如果参考节点离导体很近，湖水会更深，从而总的电容会更大。造成这一现象的原因是，导体上的电荷受到电场吸引力而被吸引至参考节点，而这种电场吸引力是由极性相反的电荷引起的。物体越近，则电场引力就越大、从而电容值越大。

当然，电容值也受到隔离两个节点的材料的影响。有些材料隔离引力的能力比其他的要强。

用于计算导体电容值的一个非常简易的模型为

$$C = \varepsilon \times A/d$$

其中，A是指定导体的表面面积，d是导体和参考节点之间的物理距离，ε是一个常数，反映了导体和参考节点之间隔离层的特性。

2.5.5　延时计算

抛开复杂的理论细节，让我们来考虑一个简单的例子，即一个反相器驱动一条导线或者导体的例子。导线等效成一个单一的电阻和一个集总电容（见图 2.20）。

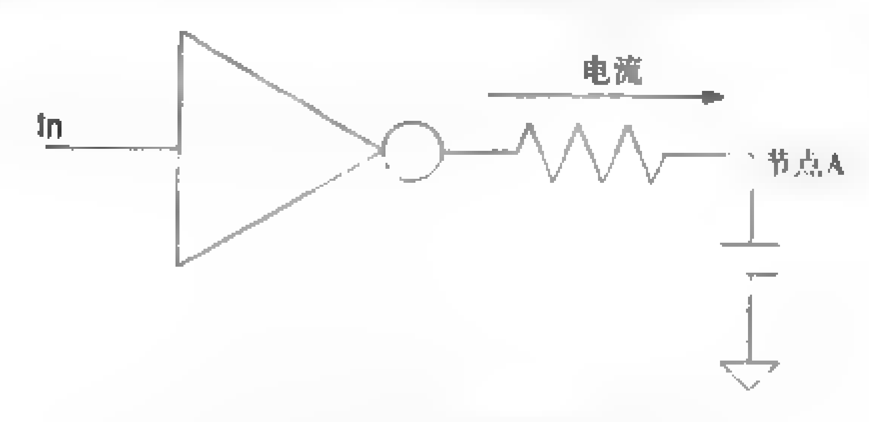


图 2.20　延时计算电路

我们的目的是计算从 In 到节点 A 的延时，总延时值取决于两个因素：

- 和反相器相关联的开关延时。反相器的延时取决于其电阻、电容的大小。这个延时通常由仿真来计算和测量，所以这里并不考虑。
- 导线的延时是由它的等效电阻和电容引起的。将导线当做一个分立元件，通过导线的延时的第一位近似值为

$$延时 = R \times C$$

这一简单的等式给出了一个分析延时的公式，利用这个公式，我们可以很容易地通过不同的连线方案来分析延时，并使我们可以在最终的版图设计中做合适的折中。

对于一个给定的电路，如果要求它的延时最小，我们需要考虑同时减小导线的电阻和电容。运用我们对电阻和电容所掌握的知识，可以通过以下的方法来优化版图设计，从而使延时最小化：

- 使导体长度最小化，这样可以同时减小电阻和电容。
- 使导体宽度最优化。减小导体的宽度可以减小导线的电容值，但这样会增加电阻值！
- 增加导体和其他参考节点的间距。这可以减小导线的电容值。通常这意味着在没有多边形和其他形状的地方布线，或者是利用顶层金属代替较低层金属。

第3章　版图设计

在第1章中，我们给出了版图设计的详细定义：

版图设计是创建工程制图的精确物理描述的过程，而这一物理描述遵守制造工艺、设计流程、通过仿真显示为可行的性能要求所带来的约束

总而言之，版图设计师是这样的一个人：通晓基础电学概念、工艺限制及特性；对空间和版图规划拥有良好的想像和直觉的能力；能够学习和使用各种各样的CAD工具

接下来，让我们来更详细地了解一下制造工艺，以及如何把版图和设计的物理描述联系起来

3.1　CMOS VLSI 制造工艺简介

设计工艺有许多种，但在本书中仅讨论CMOS工艺。首先，我们将讨论分层的制造顺序（见图3.1），而不涉及每一步骤在物理上如何实现这样的细节问题



图3.1　CMOS 制造工艺

从纯净的硅晶圆开始说起。在各个步骤之间会生长一层隔离层，用来保护在该区域中不会形成版图设计的图案。

通过用适当类型的“阱”来划分晶圆的不同区域，从而确定P基底区域和N基底区域

然后加入的是形成栅区的多晶硅。

通过多晶硅栅两旁的扩散区来确定源区和漏区。同时形成的还有其他有源区，如衬底接触和保护环。

为了使互连层连接到多晶硅和/或有源区，需要在将要进行连接的那一层之上的隔离层中创建接触孔。

互连层由金属淀积而成，同时将此前创建的接触孔填满。

最后一层称为钝化层，其上有为连线绑定的连接而创建的开孔。钝化层是一种玻璃层，它将芯片和外界隔离开来。

本框图只是关于制造工艺的一个非常简单的说明。不同工艺技术在制造步骤上会有较大的差别。

以DRAM存储器为例，它使用四层多晶硅来构建存储单元的电容。而ASIC设计仅使用一层多晶硅，但会使用更多的金属层，这些金属层是用来连接许许多多的逻辑门的。通常在生产微处理器和其他复杂ASIC设计时，需要用五至六层金属（见图3.2）。

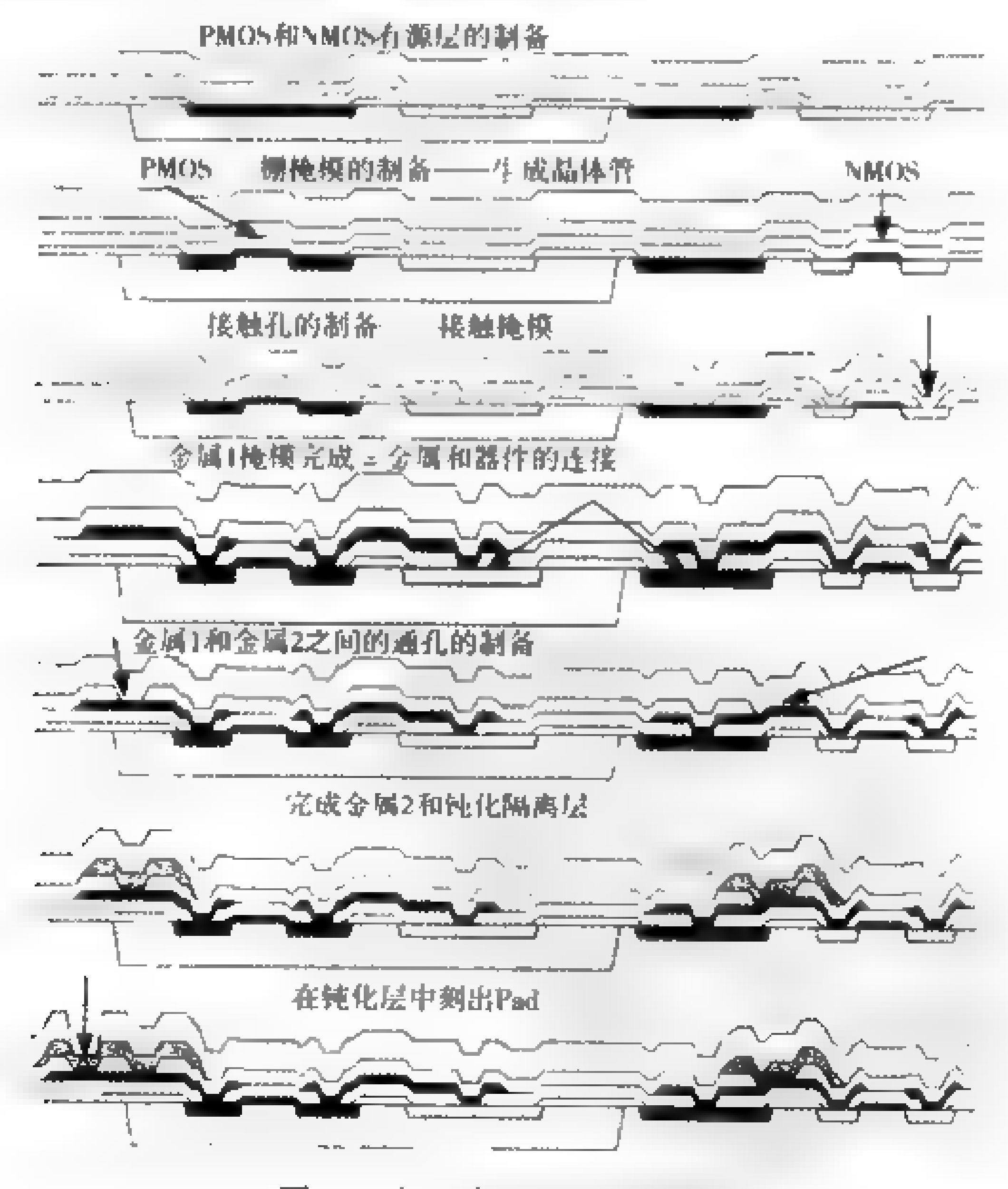


图3.2 加工步骤的横截面示例

3.2 分层和连接

让我们简单说明所用到的分层类型，并对掩模层和绘图层的概念进行介绍。

如果我们对大多数的CMOS 工艺进行分析，就会发现它有四种基本分层类型：

1. **导体**：这些层是导电层，因为它们能够传送信号电压。扩散区、金属层、多晶层以及阱层都属于此类。
2. **隔离层**：这些层是用于隔离的层，它在垂直方向和水平方向上将各个导电层互相隔离开来。无论是在垂直方向还是在水平方向上都需要进行隔离，以此来避免在个别电气节点之间产生“短路”现象。
3. **接触和通孔**：这些层用于确定绝缘层上的切口（cut）。绝缘层用于分隔导体层，并且允许上下层通过切口或“接触”孔进行连接，像金属通孔或者接触孔就是这类的例子。在钝化层上为绑定pad开孔则是接触层的另一种情况。
4. **注入层**：这些层并不明确地规定一个新的分层或者接触，而是去定制或改变已经存在的导体层的性质。例如，PMOS晶体管和NMOS晶体管的扩散区或有源区是同时被确定的。P+掩模用于创建P+注入区，它通过使用P型注入而使某一扩散区成为P型区。

将这四种类型的层结合起来使用，就可以创建晶体管器件、电阻、电容以及互连。

在几乎所有的情况下，版图设计师所需绘制的分层数目已经减小到制版工艺所要求的最小数目，这种最小数目的层称为绘图层。绘图层数目的最小化降低了CAD软件的计算需求，减少了人为错误并简化了分层的管理。

生成光学掩模的掩模层或者分层的形状有时会和绘图层不同。首先，这些掩模层的层数可能比绘图层多很多，在这种情况下，附加的掩模层是从绘图层中自动生成的。

另外，为了适应制造工艺的变化，掩模层的尺寸可能会根据绘图层做一定的调整。这个调整会由制版工艺自动完成。

应注意的是，隔离层从不需绘制出来，它总是隐含于掩模层之中而作为制造工艺的一部分。

从这点上说，任何时候提到“层”的概念，都应指绘图层。

当然，所有层的输入是用高级CAD软件完成的，而后续的分层操作同样用计算机和复杂的软件来完成。

绘制的每个图形不是以“多边形”（polygon）方式输入，就是以“线形”（path）方式输入。这两种方式之间有细微的差别，这种差别在一定程度上与计算机处理和加工版图数据库的方式有关。在某些情况下，多边形比线形更适合于版图设计，反之亦然。这些区别将在下面两节中进行说明。

3.2.1 多边形

顾名思义，多边形具有N个边的形状，这个形状在几何上有$N+1$个顶点，这就定义了该多边形的形状（计算机认定$N+1$个顶点是因为有一个顶点被重复计数了，它既被当成计数起点，又被当成计数终点）。

多边形主要用于覆盖那些无法用简单矩形覆盖的区域，如单元边界、晶体管、n阱、接触、扩散区以及晶体管栅极。另外，多边形用于定义区域的方式非常灵活，这是因为它们可用诸如90°角、45°角或在极少的情况下用手绘角度等多种角度来实现。

使用多边形的优势包括以下几点：

- 能用于圈起一个形状奇特的区域。
- 易于绘制、增加或减少。
- 在分层结构的同一级以及同一层上，易于将一多边形和其他多边形拼接起来。

参见图3.3中的一些多边形实例。

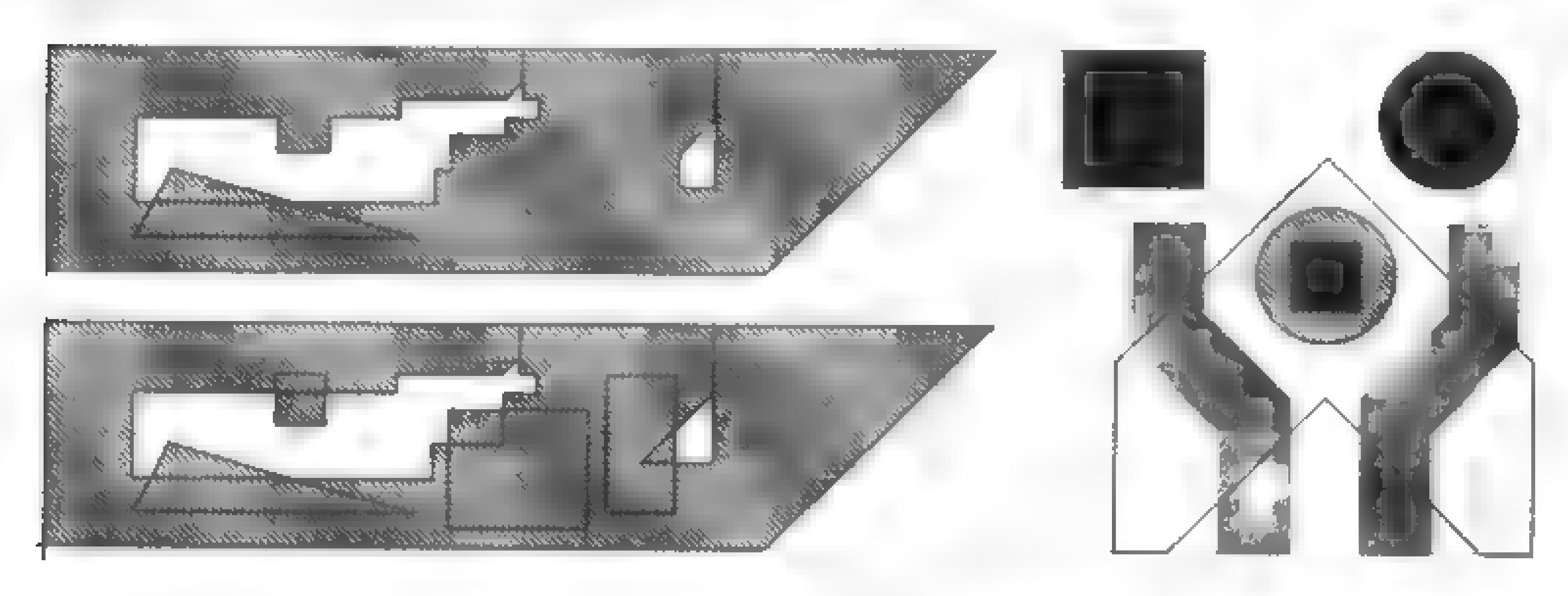


图3.3 多边形实例

使用多边形的劣势包括以下几点：

- 不易于为一致性而对复杂多边形进行修改。例如，因要求宽度一致而去修改一个多边形的所有部分将是一件乏味的事情。
- 在一些能使用线形的地方使用多边形，会比使用线形需要更多的计算机数据空间。

3.2.2 线形

顾名思义，线形是由起点、终点、中间顶点以及宽度值来确定的一种形状。由于线形具有一致的宽度，因此它主要用于连接器件，以及点对点的信号传送。

线形易于处理，并且就数据而言，线形与多边形相比使用更少的计算机资源。在定义线形时，它的顶点规定了其中心线（或边界线），再用一个附加的变量来规定其宽度就可以了。线形同样能够按90°角、45°角或者手绘角度等方式来绘制。

线形可以设计成中间对齐、左对齐或者右对齐，这意味着线形形状的顶点要么显示为居中，要么显示为偏左或偏右。

线形的终止方式是它的另一个属性。线形的长度与起点、终点有关，这一长度可能是固定的，也可能长于终点，或者以围绕终点的半圆形作为终点形状。

在实现所有这些特性的过程中，需要记住这样一些事情：目标制造工艺、CAD工具以及设计要求。图3.4给出了线形的一些实例。

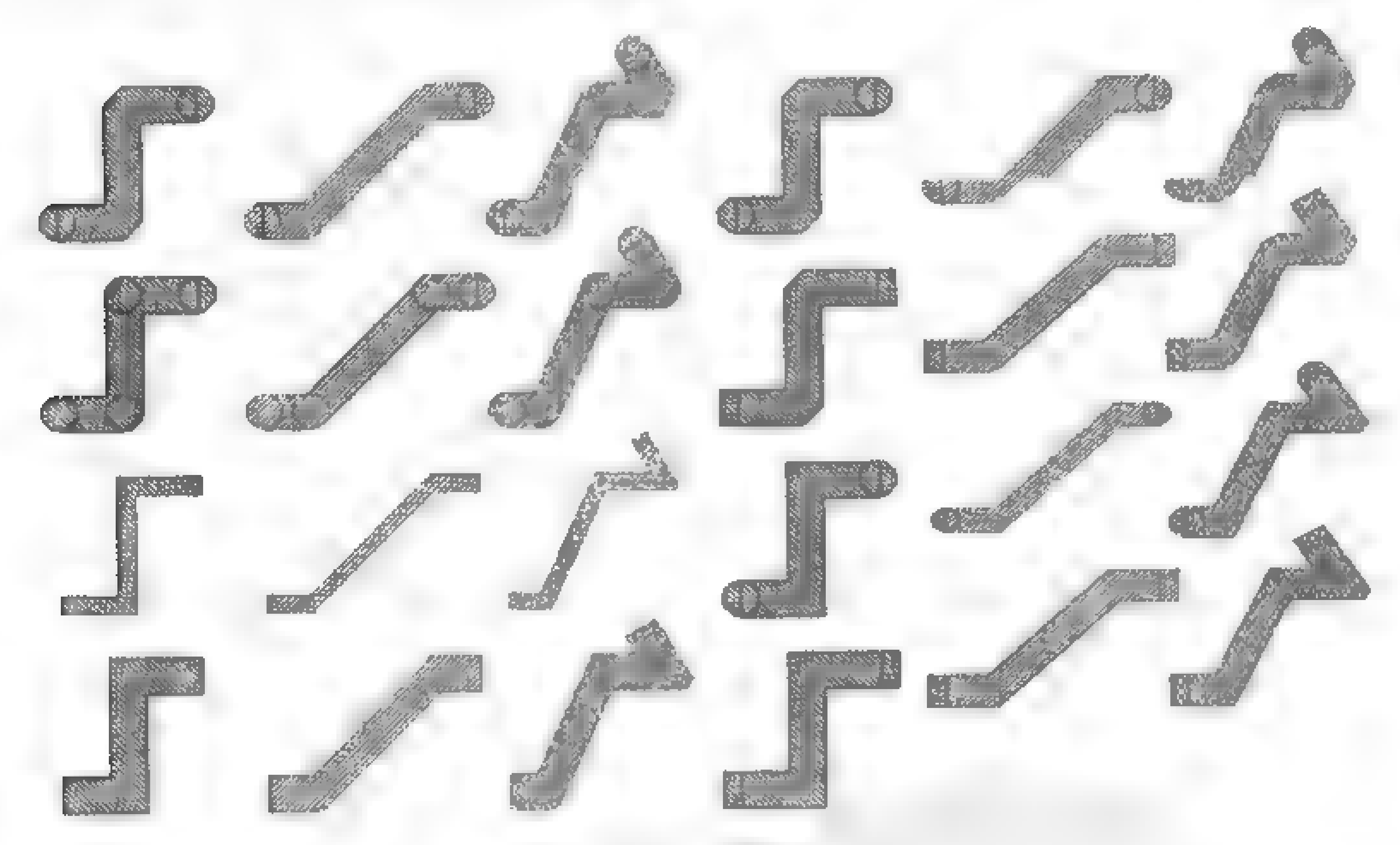


图3.4 线形实例

可以看到，线形的潜能在于：可以通过改变终端和顶点的形式来适应不同的版图风格和设计要求。

用多个线形生成版图是线形的一种有效应用方式。只要确定了所要求的形状，就可以将线形展平开来，由此得到多边形（见图3.5）。利用线形生成第一版版图会比用多边形方式更加快捷和有效。如果需要，还可以将线形转换成多边形，而逆向的转换则是非常有限的，线形并不易于由多边形来生成。

在版图设计中，线形使用得越多，版图的设计效率就越高，而能使用多少线形则取决于版图的类型以及版图设计工程师的工作习惯。线形比多边形更易于修改，并且包含更少的计算机

数据。例如，移动一段线形只需要移动一条边，而移动同样一段多边形则需要移动多边形两侧的两条边。

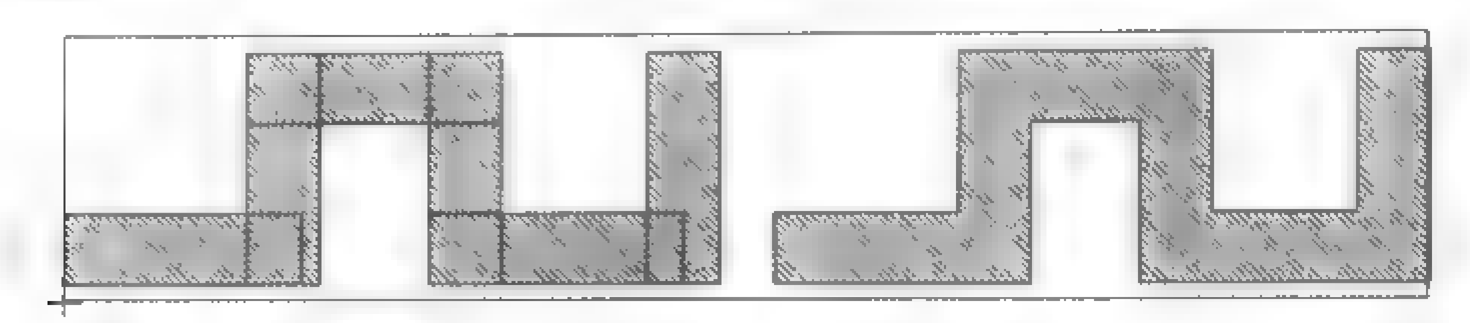


图3.5 将多个线形展平成一个多边形

长远来看，高效的工作习惯能节省时间和金钱，这是由于版图数据库的最小化也会使其他几个因素最小化：

- 存储版图数据所需的磁盘空间
- 工作站运行时内存的使用
- 屏幕刷新的时间
- 为制版工艺处理整个版图数据时，工作站所需要的CPU时间

线形的惟一缺点是，当需要将线形和多边形拼接起来时，一些CAD工具不支持这种形式的拼接。

3.3 晶体管版图简介

在开始讨论晶体管版图之前，让我们先回顾一下第2章中所叙述的电路图基础知识。图3.6的上半部分给出了PMOS晶体管和NMOS晶体管的基本符号表示，图中给出了晶体管的长度和宽度。另外，还应牢记的是基底连接也是存在的，但是为了避免将电路图弄得混乱不堪，因此将其视为隐含连接而未画出来。

在第2章中，我们说到，电流的总量取决于器件的尺寸。这就是说，当器件的宽度增加或长度减小时电流会变大。至于其原因，若能理解晶体管的物理特性取决于器件的版图这一道理，也就不难明白了。

图3.7给出了一个简单MOS晶体管的版图。注意以下几点：

- 图中给出了晶体管的所有四个终端，并将其标注出来
- 用多晶硅的多边形来确定晶体管的栅极
- 用与晶体管栅极相邻的扩散区或有源区来确定源区和漏区。应注意的是，源、漏的标号实际上可以互换。
- 这个晶体管若正巧是一个PMOS晶体管，那么通过P+注入层可使有源区成为P型掺杂区

- PMOS晶体管位于N型的阱（称为N阱）中，这就形成了晶体管的基底节点
- N型有源区（没有P+注入层）会与N阱相连通，这是因为N阱和N型有源区具有相同的掺杂类型（N型）
- 源、漏、阱的连接是通过另外的接触层来实现的 这类接触层中最典型的就是用于连接第一层金属层的接触层。
- 将宽度和长度正确地标注出来，注意宽度大于长度！

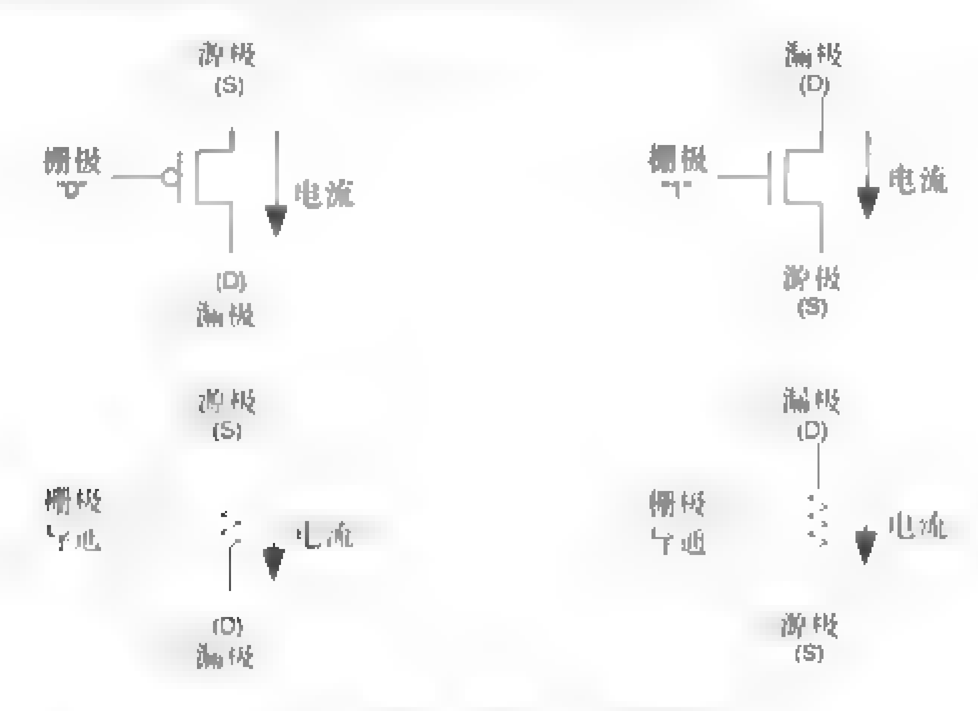


图3.6 PMOS和NMOS晶体管

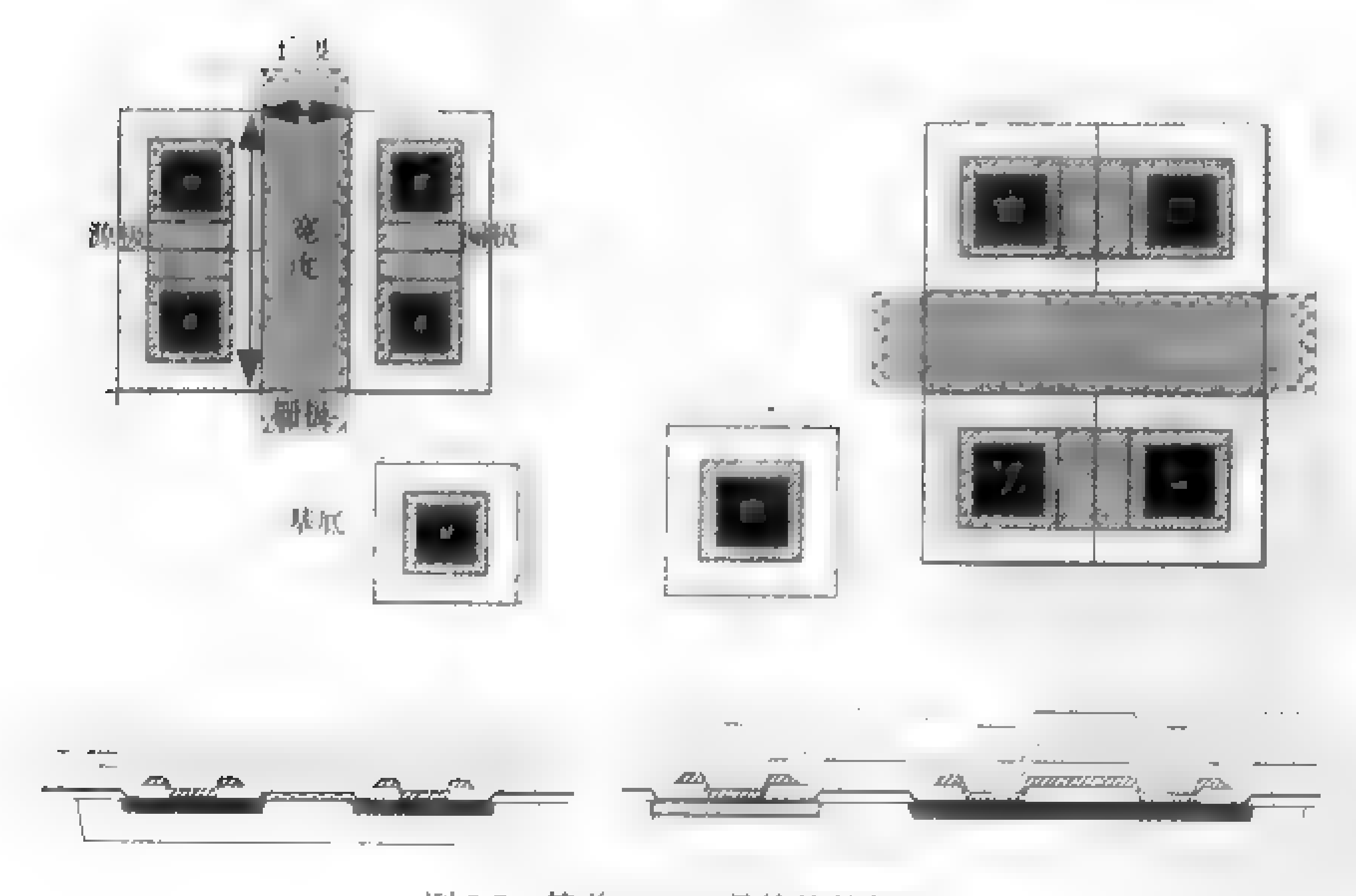


图3.7 简单PMOS晶体管的版图

晶体管的宽度和长度是我们必须完全掌握的两个最重要的尺寸参数。

如前所述，当业界人士谈论特定工艺下栅的尺寸时，他们是指栅的最小长度。请注意以下几点：

- 就版图而言，晶体管的长度是源极和漏极之间的距离。把这段距离称为长度也许并不直观，因为晶体管长度的物理尺寸要比其宽度小。接下来的段落会说明这一约定的形成原因
- 就晶体管性能而言，晶体管的长度是这样定义的：当栅极"导通"或"开启"时，为了形成一个可测的电流，电子所必须移动的距离即为晶体管的长度。应记住的是，栅极电压控制电流。如果缩短源极和漏极之间的距离，则栅极电压对电流的控制能力就会增强。概括地说，在同一加工工艺上，如果两个晶体管宽度相同，而长度不同，则栅长度短的晶体管必会产生更大的电流。更大的电流在概念上则意味着更快的性能。
- 就制造能力而言，晶体管的长度是多晶硅（多晶）能够可靠制造的最窄可能长度。多晶尺寸越小，晶体管就越小，从而IC就越小。所以，使用最小的栅长度来使芯片面积最小化是极具吸引力的。

让我们来研究一下晶体管的宽度。

可以将晶体管宽度想像为一系列平行的沟道，电流通过这些沟道从源极流向漏极。宽度更宽的晶体管有更多的有效沟道；更多的沟道则意味着更大的电流。

再一次比较两个晶体管，这一次两个晶体管栅长度相同而宽度不同，栅宽度更大的晶体管会产生更大的电流。

为了帮助读者记忆晶体管长度和宽度的约定，我们可以将晶体管想像成一座桥，桥的长度是河岸之间的距离，而宽度则是桥上能容纳的车道的数量。过桥的交通量受桥的长度和宽度限制，同理，电流受晶体管长度和宽度的限制。

如果桥设计成允许在1分钟内通过100辆车，那么桥必须造得足够宽才能达到这一目的。在多数情况下，桥长是固定的（类似于允许的最小栅长），要达到我们的目标，我们惟一可以做的就是调整其宽度。

最后一种需要考虑的情形是在某些情况中，我们可能需要一个慢速或者弱晶体管！通过使晶体管宽度最小化和/或增加栅长度，就能够很容易实现这一要求。延时元件或者弱反馈器件是我们使用慢速晶体管的实例。在这类情形中，栅的长度确实大于其宽度（见图3.8）。

应记住的是，晶体管的长度和宽度之间的区别，以及如何将它应用于晶体管的版图！

为了完整地阐述这些概念，图3.9给出了一个NMOS晶体管的版图。

- NMOS晶体管并不位于任一类型的阱里，它位于纯净的衬底上。在这种情况下，可以推断衬底是P型的。衬底形成晶体管的基底节点。
- P型有源区（用P+注入层）形成和衬底的连接，这是因为衬底和有源区是同一种掺杂类型（P型）的。

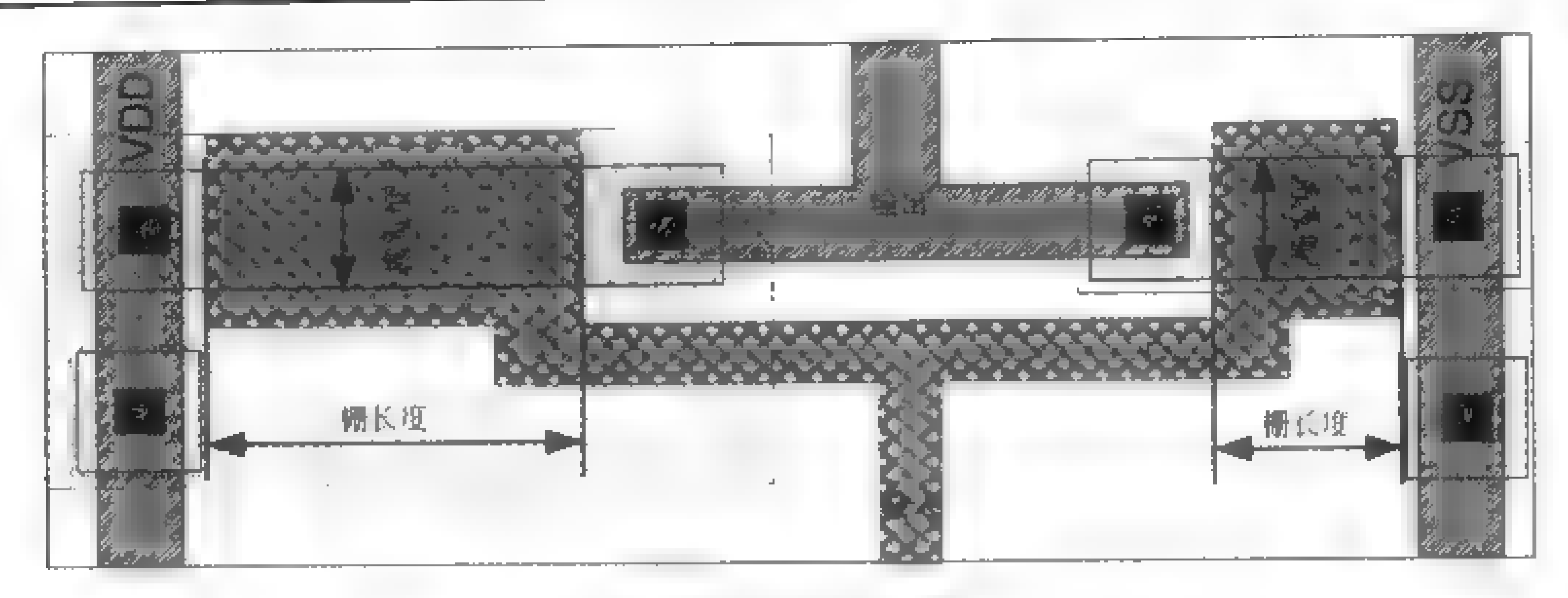


图3.8 弱反馈反相器

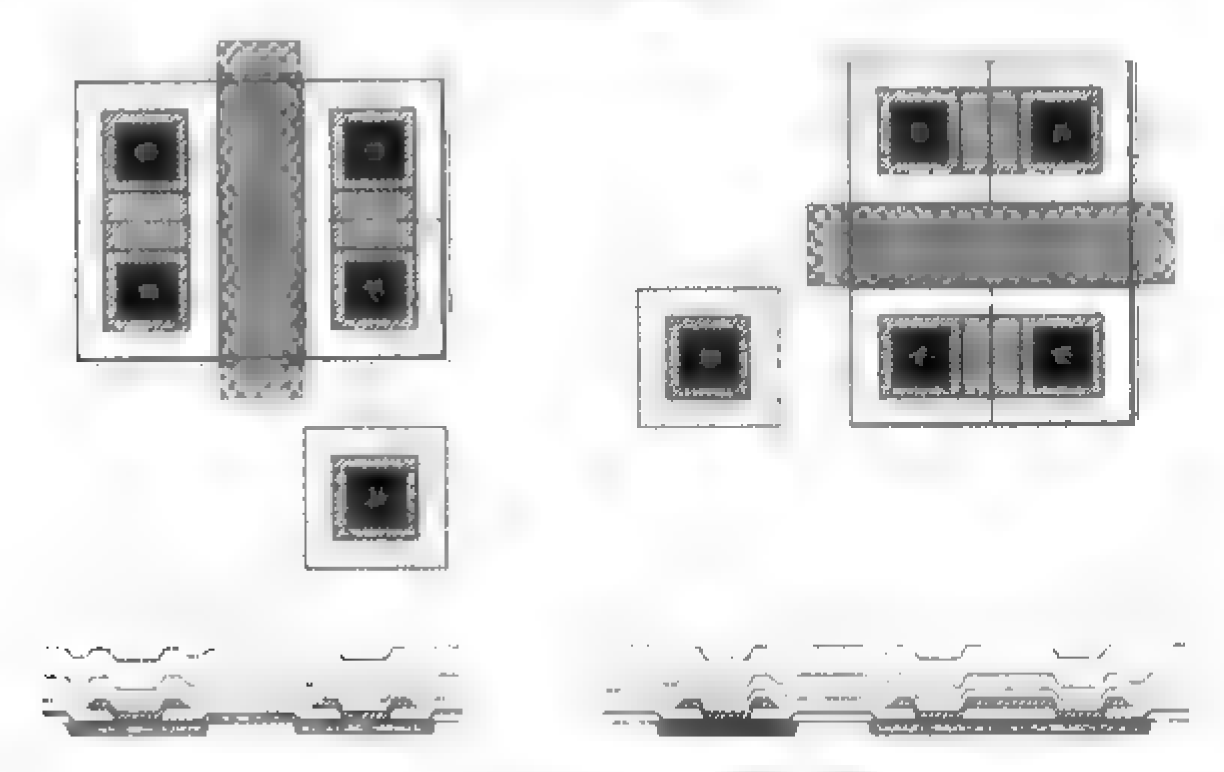


图3.9 简单NMOS晶体管的版图

3.3.1 基底连接

知道了两种基本类型的晶体管的工作原理及其外观后，接下来让我们回顾一下基底连接节点，看一看它是如何实现连接的。通过理解晶圆和晶体管的横截面，能使我们毫不费力地理解这一问题。但应注意的是，对版图设计工程师而言，只能理解基底连接这一概念，而不能影响它的设计。

当前多数（但不是全部）未加工的硅晶圆是P型的，因此，在两种类型的晶体管中，NMOS晶体管更易于设计，在纯净的衬底上可以很简单地实现其版图（见图3.10的右侧）。

为了生成PMOS晶体管，我们需要创建一个单独的基底节点，这需要通过另一版图分层来实现。这一分层通常称为N阱；当其实现时，它就形成一个N型衬底岛。在此N阱里实现

P型有源区就可以得到一个PMOS管，而此PMOS管的基底连接由N阱确定（见图3.10的中段）。

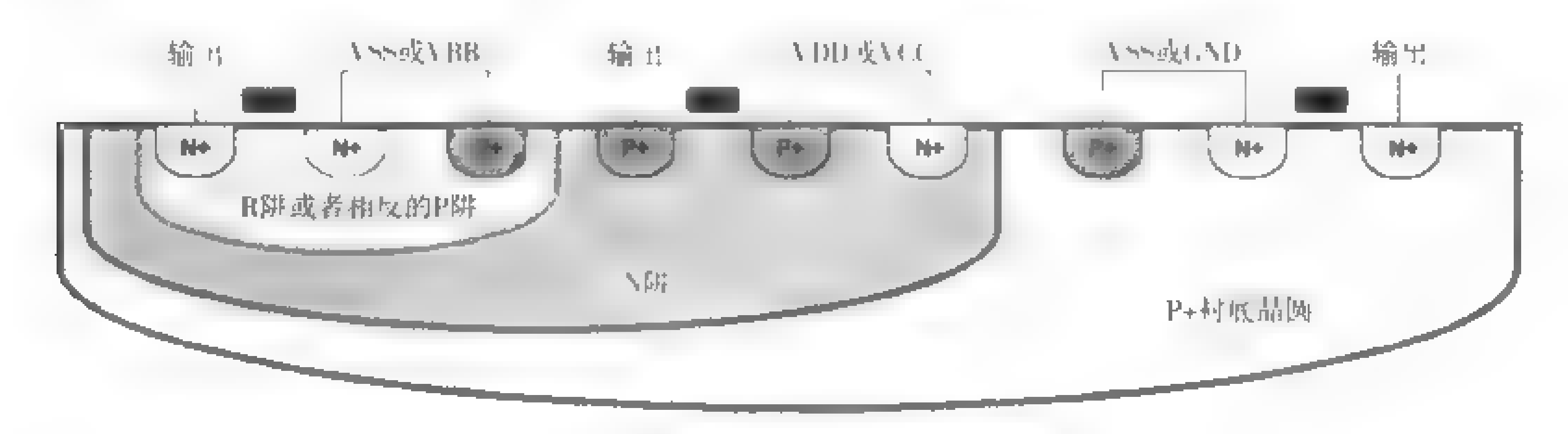
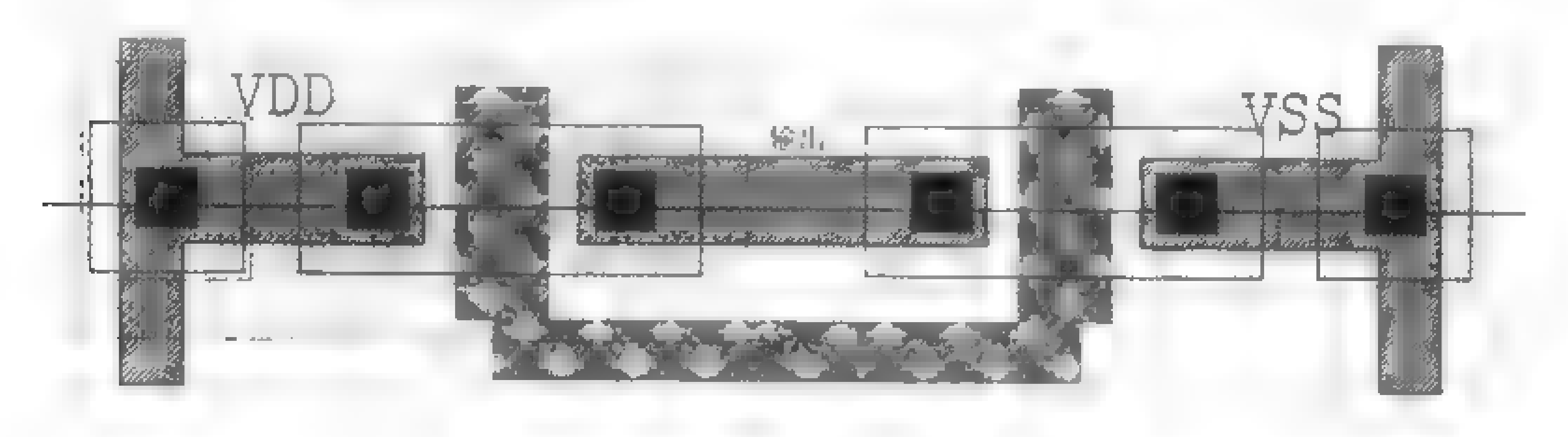


图3.10　显示衬底连接的晶圆横截面

图3.10的左侧给出了一种NMOS晶体管的设计，它有和衬底不同的基底节点。在N阱里实现一个相反的阱（R阱）或者P型阱（P阱）。此区域为晶体管在它的内部实现创建了一个单独的P型基底节点。这是一个DRAM工艺中的衬底连接的实例。

在N型晶圆的情形中，晶体管连接的极性和前面给出的极性是相反的。图3.11给出了在ASIC工艺中一个反相器的衬底连接。



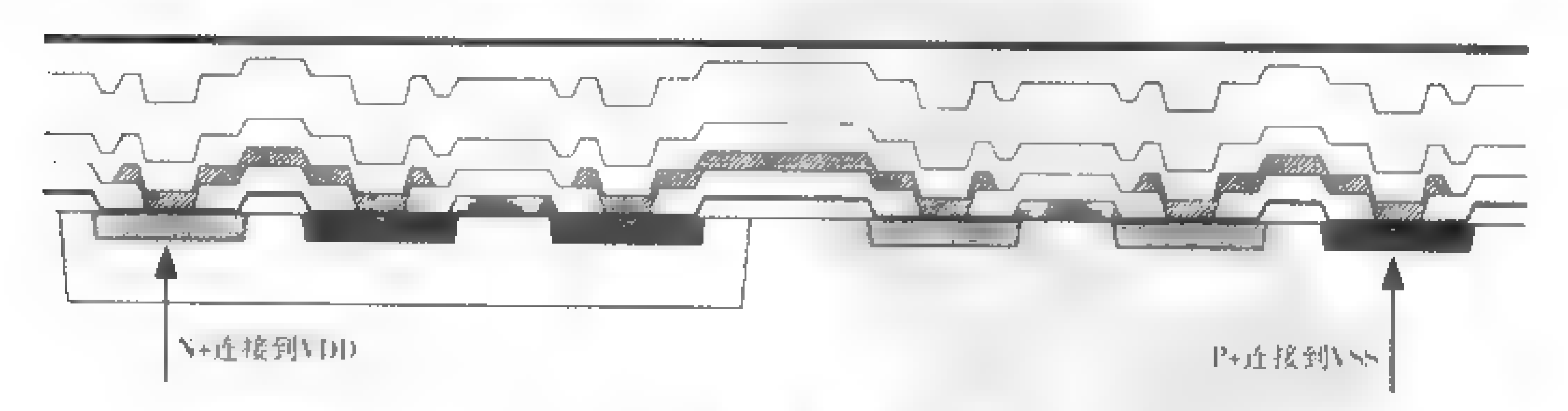


图3.11　反相器基底连接的横截面

3.3.2 导体和接触孔

从版图设计的角度来看，导体和接触孔是简明易懂的。让我们再从制造的角度来了解一下接触孔的构造，这样我们就能够像版图设计工程师一样理解它们的应用和局限了。

对于不同的工艺而言，其工艺的定义也是完全不同的。一个典型ASIC工艺只有一种类型的多晶硅用于栅，而有两到四种类型的金属层用于互连。高级ASIC工艺可以拥有高达六层的金属层用于互连，并且将称为金属0的低层金属用于源/漏的连接。对于DRAM存储器制造，当今典型的工艺是拥有四种类型的多晶硅，以及三到五层金属用于互连。就以上任何一种情形而论，对工艺的导体层的定义都非常复杂。

在业界，接触孔（contact）和通孔（via）这两个名词有着细微的差别。接触孔特指最低层金属孔，用于将最低层金属和多晶硅或者扩散层连接起来。而通孔则是指允许更高层金属进行相互连接的孔（如金属1到金属2，金属2到金属3）。

关于接触孔和通孔的说明，请参考图3.12。

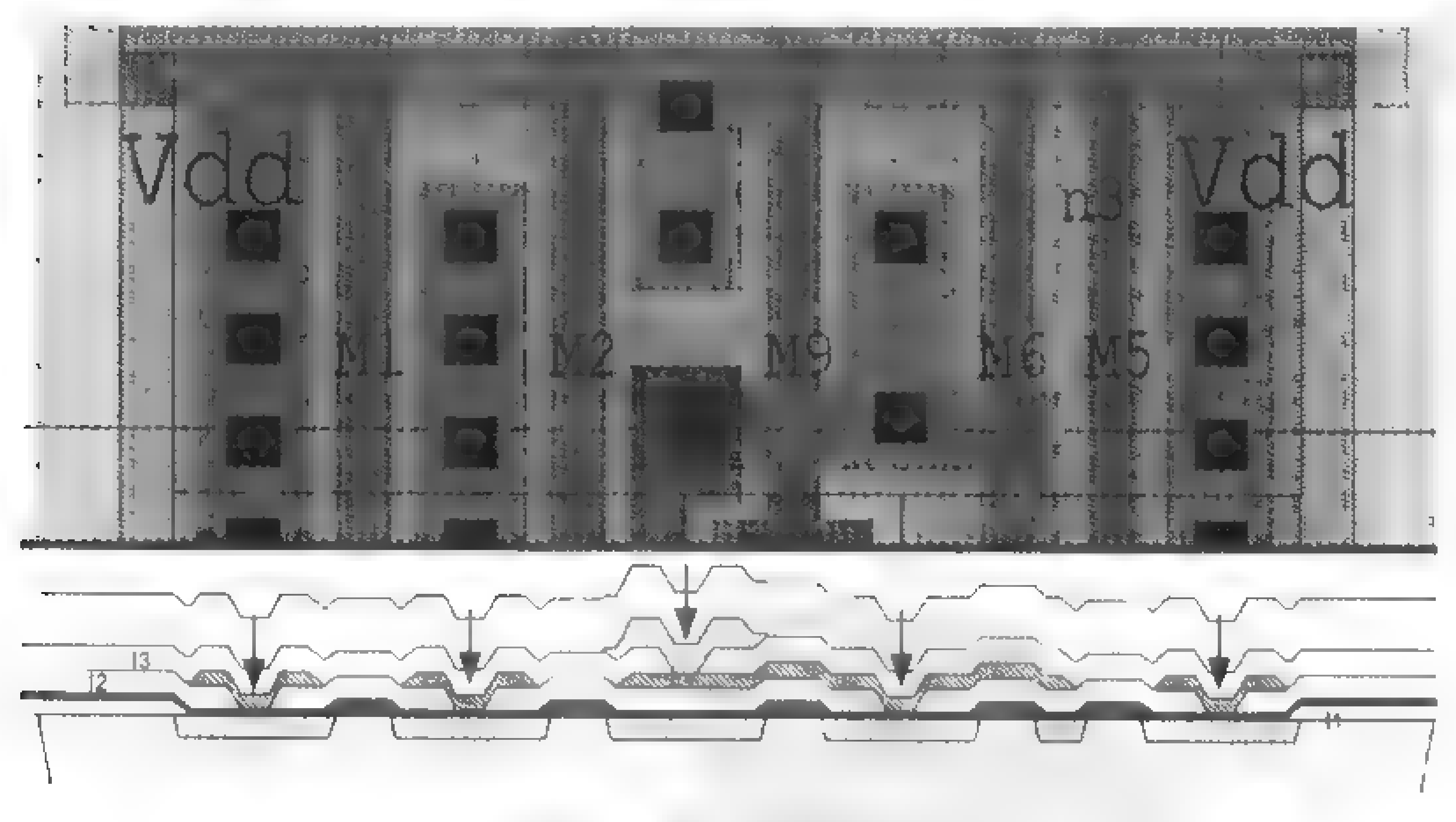


图3.12 接触孔和通孔的说明

正如从图3.12所给出的横截面中看到的那样，在各种导体层之间有不同的隔离物。I1是扩散区和多晶硅之间的隔离物，I2是扩散区和金属1之间的隔离物。隔离物上的孔生成穿过金属1到更低的有源层/漏层之间的“接触孔”。I3是金属1和金属2之间的隔离物，在它上面的孔即为通孔。

在多数情况下，要考虑到“接触孔”和“通孔”之间应有一段距离，但是在现代的多数工艺中，通孔能被放置在接触孔之上。在一些看重芯片尺寸（成本）的复杂工艺中，允许所有通孔从上到下排成直线，这称为“层叠式”通孔工艺。

根据电阻（R）、电容（C）以及拓扑要求，每层金属都有不同的特性。还需考虑的是，工艺中较高层的金属层需要更多的通孔来连接下方的晶体管分层。而这些通孔会增加电阻。我们将在稍后分析如何处理并利用这些工艺的电气特性。

3.3.3 反相器版图

我们已经学习了所有有关晶体管版图设计的基本概念。接下来，我们将再次关注最简单的晶体管组合——反相器（见图3.13）。

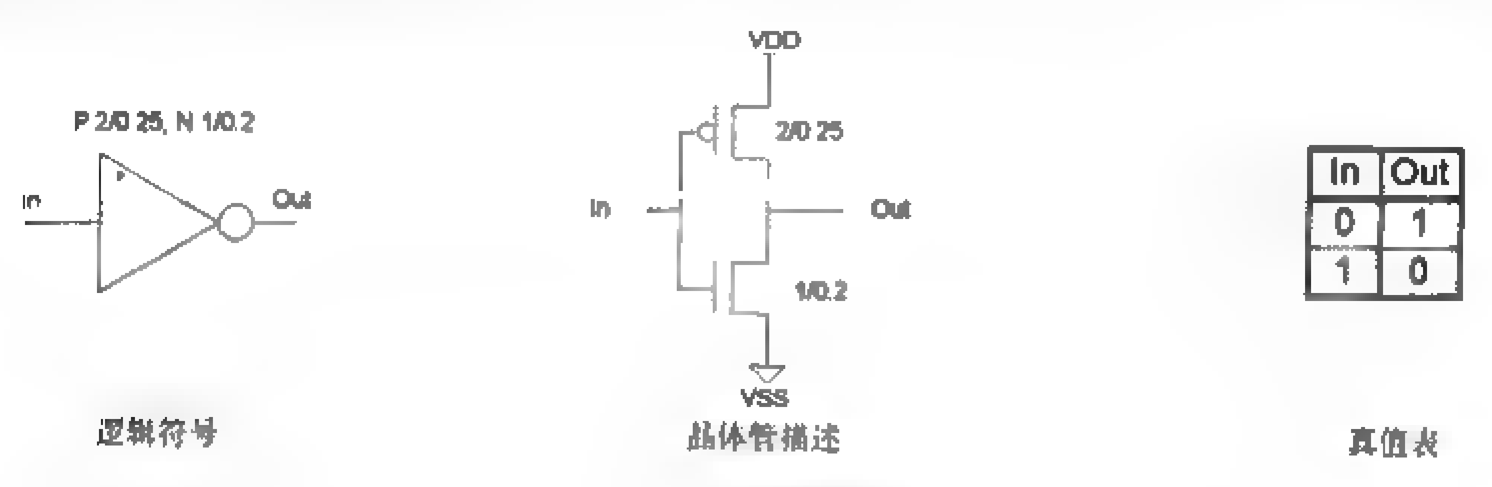


图3.13 反相器描述

正如我们所看到的那样，反相器的晶体管描述非常简单。既然我们已经了解了晶体管的外观形式和连接关系，那么就能够生成反相器的版图。通过分析图3.14，我们可以观察到：

- 无论在电路图中还是在版图中，PMOS晶体管都与VDD相连接。
- 在电路图和版图中，NMOS晶体管都与VSS相连接。
- 在电路图和版图中，NMOS晶体管和PMOS晶体管的栅极上有相同的IN信号，而其漏极上有相同的OUT信号。
- 两种晶体管的宽度不同——在此例中，PMOS晶体管的宽度是NMOS晶体管的两倍。
- 两种晶体管的长度看似相同，但却不同，我们很难辨别它们的差异。
- 对于N阱来说，N+区域实际上是与VDD相连接的，而电路图中没有显示这一连接关系。
- 对于衬底来说，P+区域实际上是与VSS相连接的。而电路图中没有显示这一连接关系。

在后面的章节中，我们还将介绍反相器的设计，在此给出反相器是为了使我们对一个完整版图单元的外观有所了解。

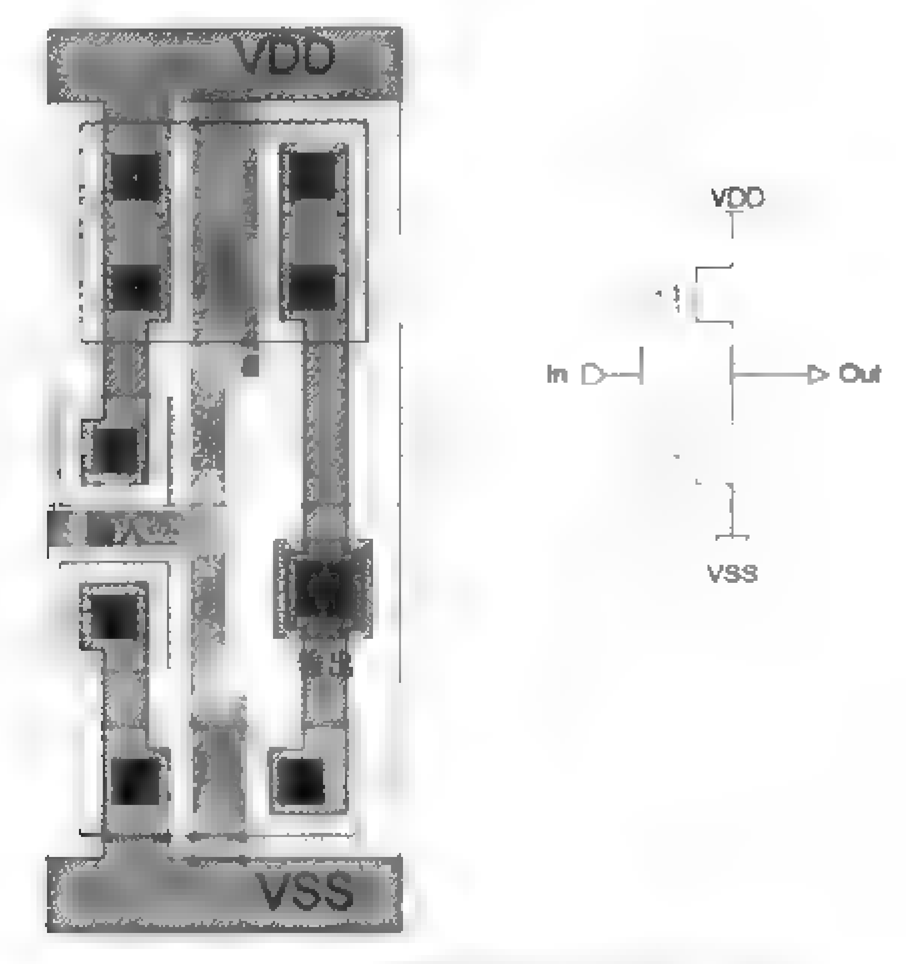


图3.14 反相器版图和晶体管级电路图

3.4 工艺设计规则

设计规则是指，在对给定设计进行版图设计时所必须遵守的一系列规则。对于所有我们已经介绍过的组元（多边形和线形，晶体管以及接触孔）都有其相应的设计规则。从根本上来说，这些设计规则体现了制造工艺的物理限制。

在那些有集成电路芯片制造能力的企业中，会有一批人专门负责定义和优化制造工艺。此"工艺"组定义设计规则时必须既要考虑到制造成本、成品率等问题，又要考虑到能由生产设备和加工步骤制造的最小特征尺寸。其他可能影响设计规则定义的因素还有：制造工具和工艺的成熟度，IC的市场需求或者代工服务。

总之，设计规则就是用来帮助版图设计工程师理解、说明物理三维空间的限制以及CAD和版图工具环境中的制造公差的。

3.4.1 宽度规则

多边形的最小宽度（在制版时，所有的线形都会转换为多边形）是一关键尺寸，它定义了制造工艺的极限尺寸（见图3.15）。晶体管的最小栅长就是这一规则的典型例子。

如果在某一分层中违反了最小宽度规则，那么就可能在该层上产生开路现象。如果宽度小于某一特定值，那么制造工艺就无法保证可靠地制造连续的连接或连线。因而在线形中的某一点若违反了规则，那么在这个点上就很可能会产生裂口。

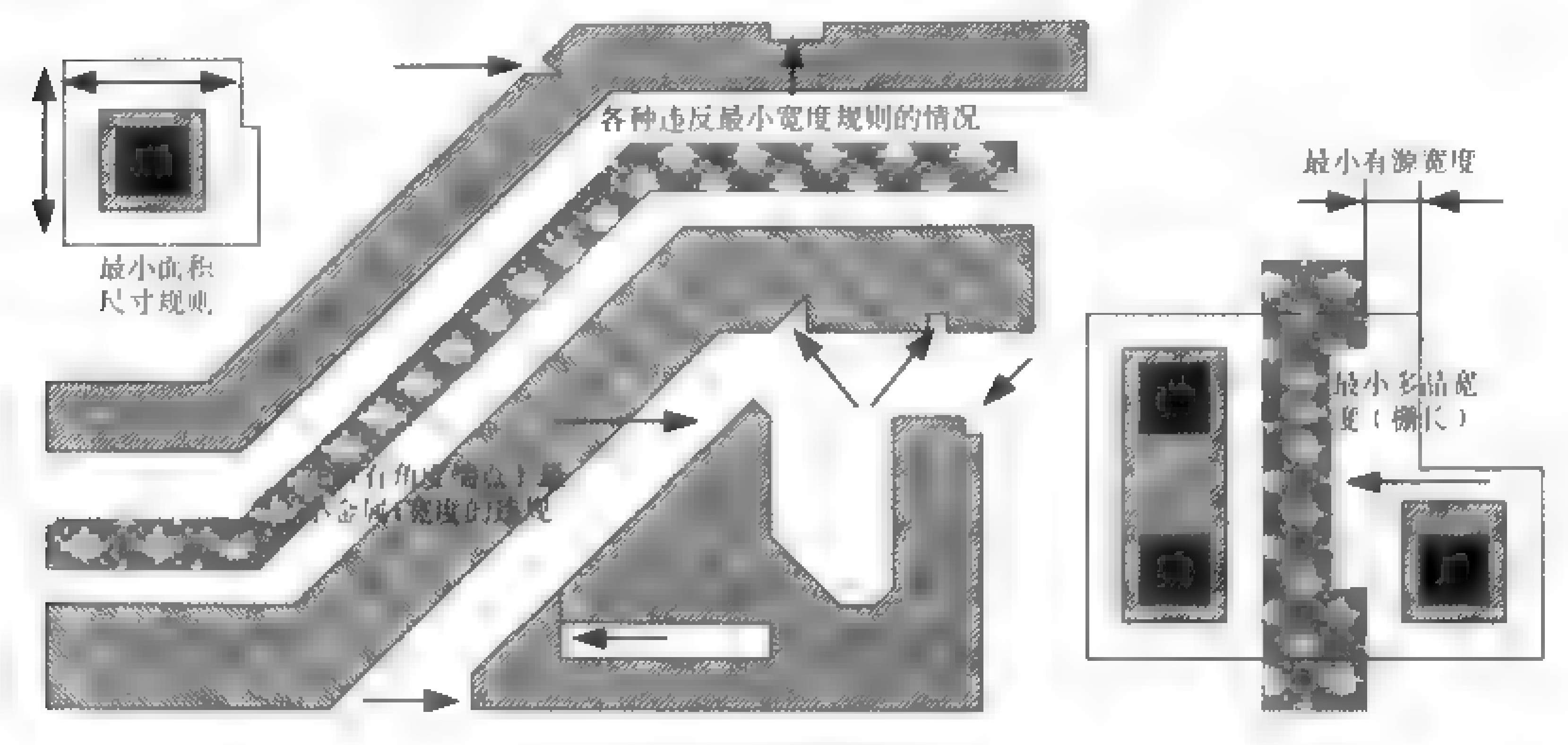


图3.15 宽度规则示例

除单个多边形之外，宽度规则还可以应用于诸如晶体管之类的结构，或者带有电气特性或其他特殊特性的单个多边形上。与电源相连的金属层就是一个带有特殊电学特性的多边形的实例。由于这些金属层上通过的电流较大，这就迫使其宽度要比最小设计规则大得多，这一宽度的确切值由电流的大小决定，而不会指定为固定值。当大电流穿过窄的金属线时，金属线就会像熔丝一样，通电时间过长或维持在较大电流的峰值时，都会使金属多边形在应力作用下断开。

多边形的长度通常没有限制；可是在某些工艺中，可能会对最小面积进行规定（例如，对于通孔或接触孔来说，必须同时满足宽度和长度规则）。为了弄清楚这一点，请参见图3.15给出的例子。

3.4.2 间距规则

间距规则（space rule）是另一个关键尺寸，它指的是两个多边形之间的最小距离。一般来说，间距规则可以用来避免在两个多边形之间形成短路。

在某一分层上，间距规则和宽度规则一起，共同定义该分层的间距（pitch）。在考虑内部互连和布线空隙率时，分层的间距是一个重要参数。将金属线数目乘以分层间距，可很容易计算出布 n 条金属线所要消耗的布线面积。请参考本书附带光盘中有关计算间距的实例。

图3.16说明了以下几点：

- 1和2是在平行和对角情况下，金属1之间的最小间距规则检查示例
- 3是多边形以45°角并行的情况下，多晶之间的间距规则示例
- 4、5、6是多边形成90°角和45°角的情况下，金属2之间的间距规则示例。
- 7是在只有单一方向上的距离或者拐角情况下，有源之间的间距规则检查示例。
- 8是例3的一种例外——多晶硅多边形之间的间距规则取决于它们所处的位置。典型的例子为：晶体管结构内的多晶硅栅之间的间距值和晶体管结构外的多晶硅栅之间的间距值是不同的

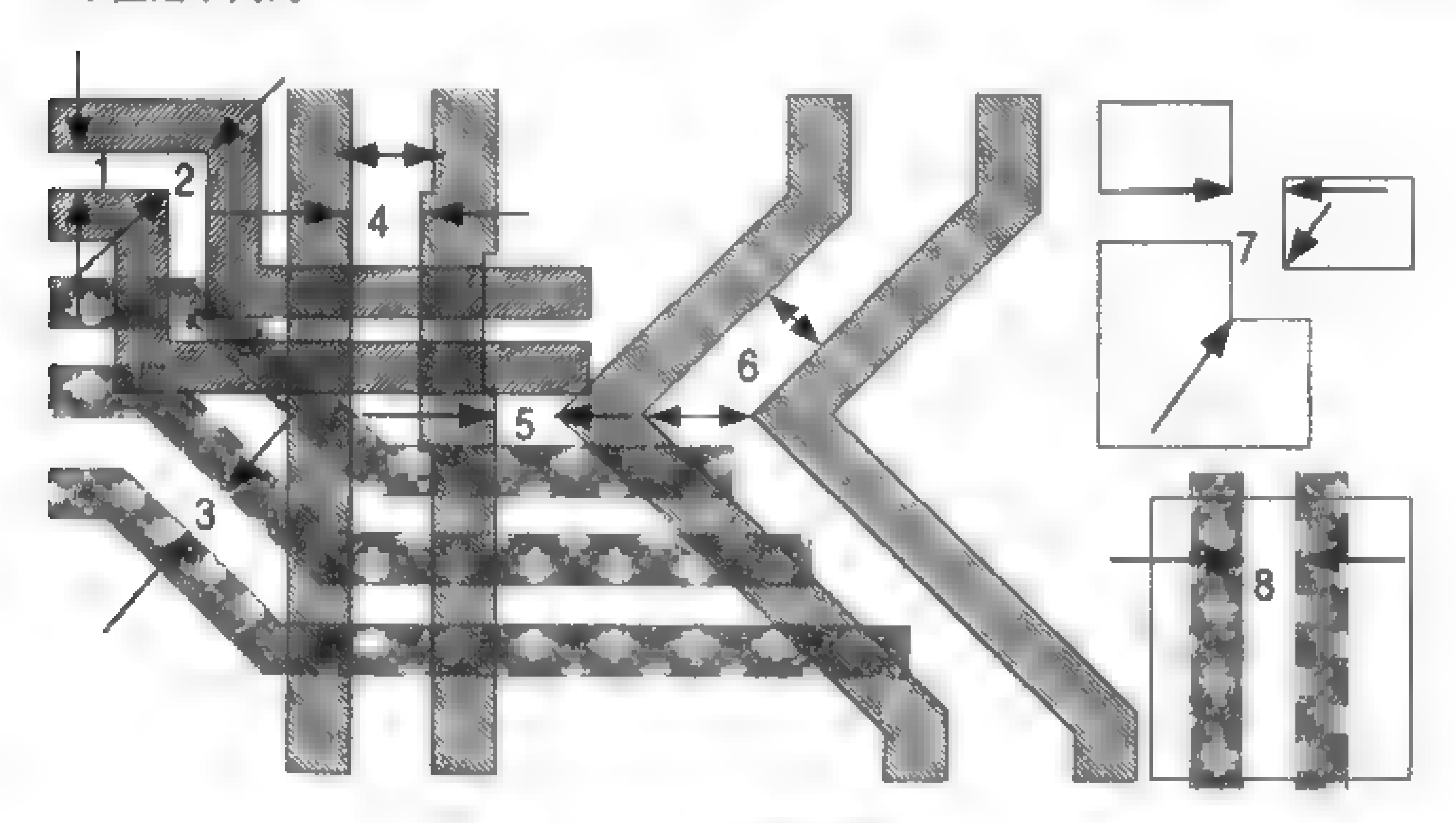


图3.16 间距规则示例

与宽度规则类似，间距规则不但应用于同一层上的多边形，也应用于不同层之间或不同情况下的多边形或结构。有源上的接触孔和多晶硅栅之间要求有一定的间距，这就是一个不同层之间遵守间距规则的实例。暴露在外的Pad电路与内部敏感电路之间应该保持一定的距离则是不同结构之间运用间距规则的例子，这一间距确保了在任何条件下芯片操作都是可靠且稳定的。

通过对工艺横截面图的观察，我们可以很容易地理解在一组设计规则中定义的许多间距规则。这一点将在3.5节中的纵向连接图中得到充分的说明。

在图3.17中，我们可以看到，两个晶体管中的多晶硅栅和接触孔之间的间距是不同的。而通过观察横截面图，我们首先要注意的是两个晶体管的源区和漏区并不相同。

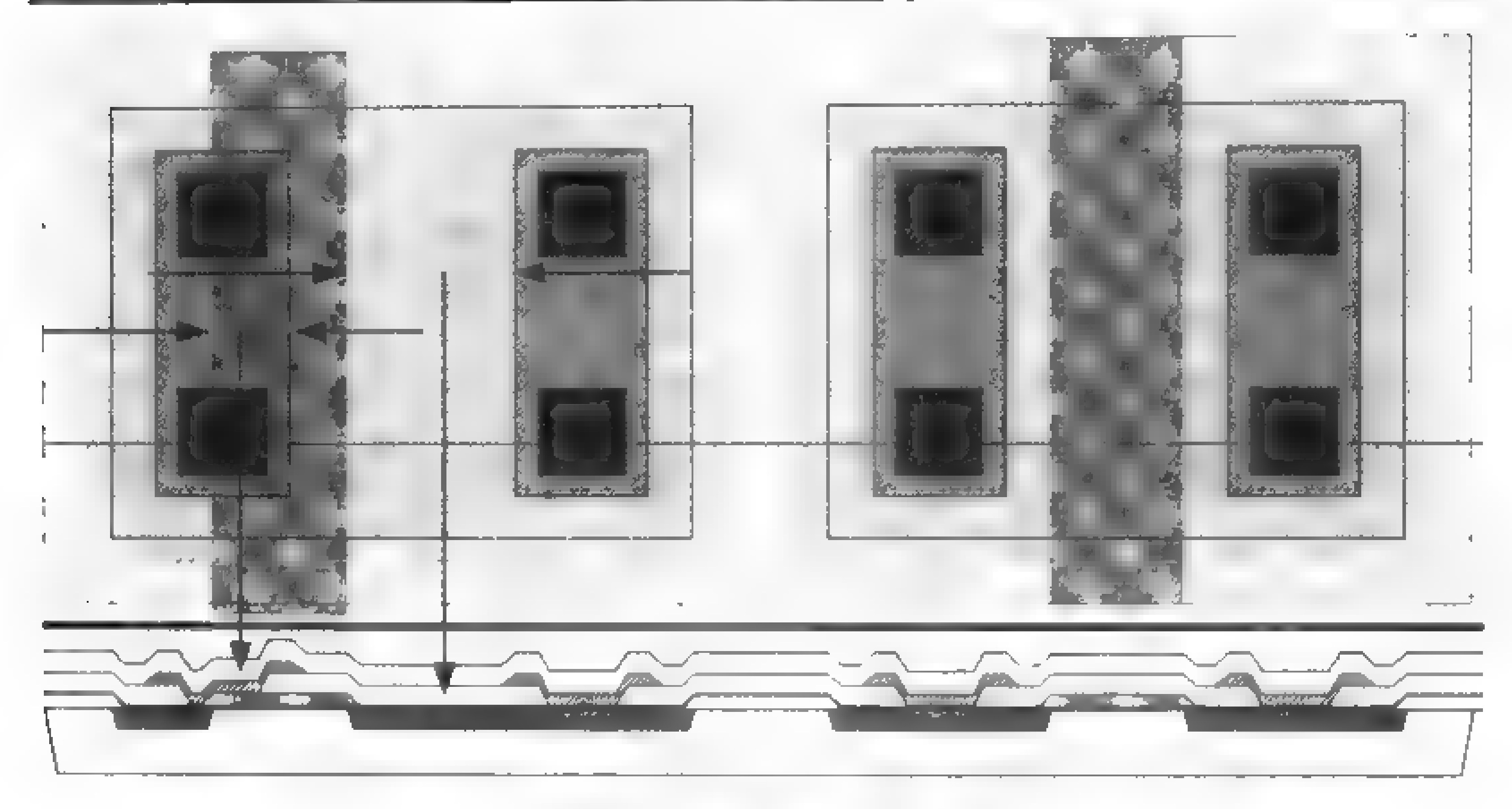


图3.17 间距规则的另一示例

但更重要的是，左侧的晶体管将多晶硅栅直接放置在接触孔的下方，它违反了接触孔与多晶硅栅之间的间距规则，这会造成金属和多晶硅栅之间的短路现象。这个问题在三维空间视图中可以很容易地观察到：图中横截面切割线位于较低层接触孔的中间位置。

3.4.3 交叠规则

顾名思义，交叠规则（overlap rule）被定义为一个多边形与另一个多边形之间相交叠或相包裹的最小尺寸限制。金属层与通孔或接触孔交叠就是这一规则的一个典型例子。

应注意的是，该规则中所指的多边形总是位于不同层上，而这也正是版图设计中需要此类规则的主要原因。只要是使用不同层上的多边形来制造某种结构，那么放置多边形的预期位置与实际位置之间就很可能会出现偏差。对于某些分层来说，多边形间的偏差可能会导致电路连接出现不希望有的开路或短路。从根本上来说，交叠规则通过确保预期的连接关系不会因制造工艺而遭破坏，来减少制造工艺中由于分层间细微的偏差所带来的影响。

让我们来考虑这样一种情况：在两个互连层之间有一个接触孔。在物理上，该接触孔多边形会在两个互连层之间的隔离层中形成一个孔，而上层金属必须填充这个孔，从而与下层相接触，以完成相应的连接。

交叠规则指出上述的两个层之间仅相互交叠是不够的，其中一层必须以一定值包裹着另一层。这个值就是交叠规则所要求的值。以接触孔为例，上下两层必须完全覆盖接触孔，并且以

交叠规则值来包裹该接触孔。如果其中有一层没有充分地覆盖和包裹接触孔，那么就不能在所有制造环境下确保连接的可靠性。

满足交叠规则有哪些益处呢？在上述例子中，如果只是多边形的实际位置与预期位置之间没有很好地对准，那么，仍然会有足够的上一层材料可以用来填充接触孔。但是，如果上层或下层金属没有完全覆盖接触孔，那么可用于电气连接的区域就会相应地减少，这样将导致连接不充分或者不稳固。

在图3.18的例1和例2中，可以看到有源与金属1间不充分连接所造成的后果。如果有源没有完全覆盖接触孔多边形，那么接触孔底部就不够宽。如果金属没有完全覆盖接触孔多边形，那么接触孔就不会被完全填充。在这种情况下，接触孔会使得可用于连接的表面区域更小。第三种情况中，金属1和金属2之间存在交叠问题。通孔也没有和金属1交叠，所以即使有连接，它也是最弱的。

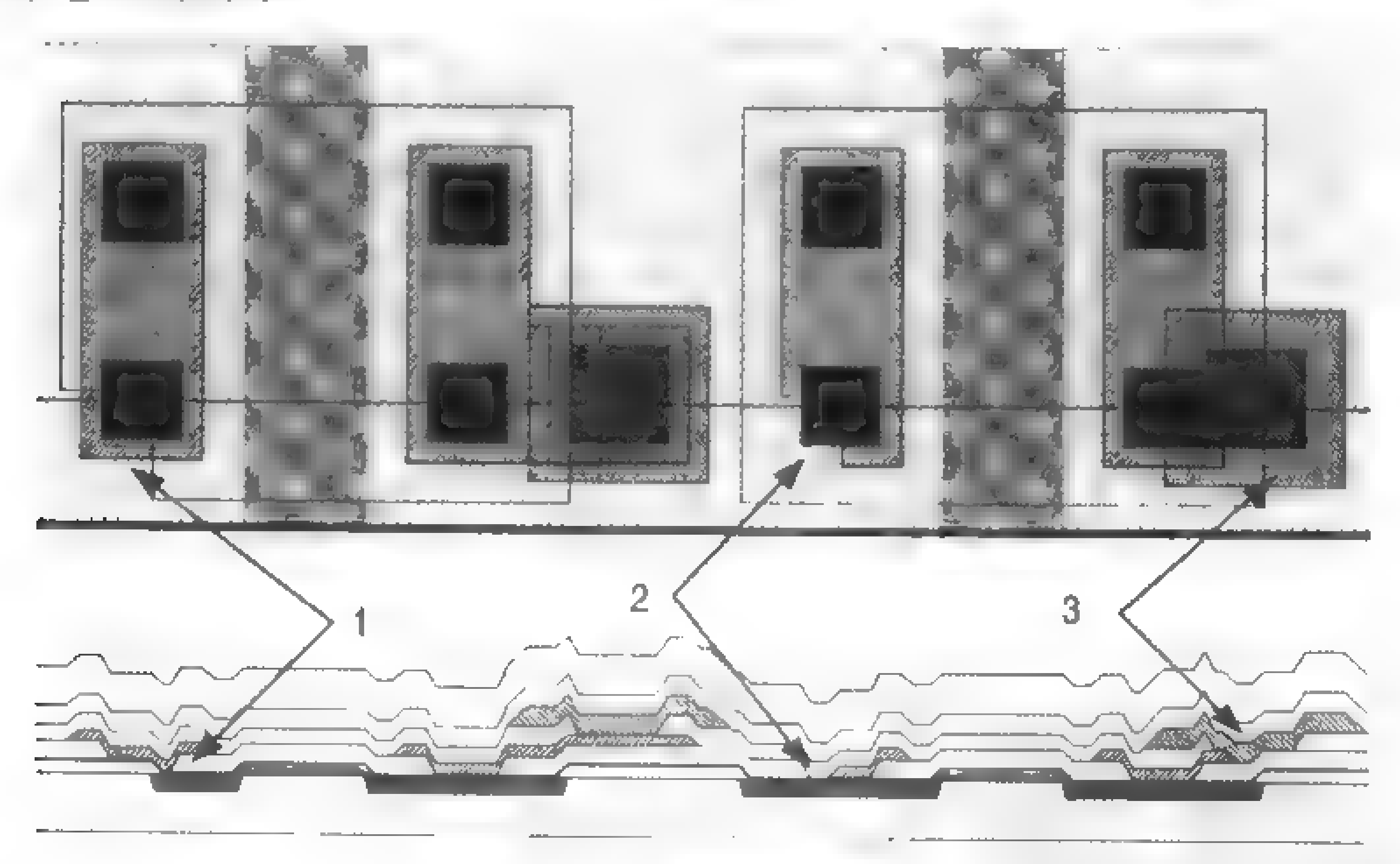


图3.18　交叠规则示例

图中的例子说明了在什么情况下更可能出现电路开路的现象。现在，让我们看下一个例子，在这个例子中，由于违反了交叠规则，产生了短路现象。

在这个例子中，让我们考虑一下多晶硅栅和有源层相结合的几种不同的晶体管版图。如果既要确保晶体管的尺寸精确，又要避免源极节点和漏极节点之间发生短路现象，那么就需要遵守一些与晶体管相关的特殊交叠规则。

通常，有这样两种交叠规则：有源交叠栅，栅交叠有源区。图3.19给出了四种不同情况

图3.19　另一些交叠规则示例

Out4节点是有源多边形充分交叠栅层的例子。Out4节点被很好地定义了。与之相对比的是Out3节点。Out3节点中的一个过薄区域在制造中很可能无法生成。

Out1和Out2是栅层交叠有源层的例子。可以看出，由于栅层没有充分交叠有源区，使得节点Out2和VDD间为有源多边形的一部分而短路。

3.5　纵向连接图

在大多数ASIC工艺中，对于连接关系有很多简单易懂的规则。ASIC工艺中会有一层用于栅的多晶以及二到六层金属，这意味着我们只需要处理一种接触孔以及两到五种通孔就可以了。这样的连接关系是很容易掌握的，因为若要连接有源和金属6，则一定需要用所有可能的接触孔和通孔。

在更加复杂的工艺中，建议生成一个关于层连接关系的纵向图表，这样可以使版图设计工程师更全面地掌握各种连接关系。图3.20给出了DRAM工艺，它由四层多晶和三层金属构成。在这种工艺类型中，对分层的使用以及连接规则通常会有相应的约束。

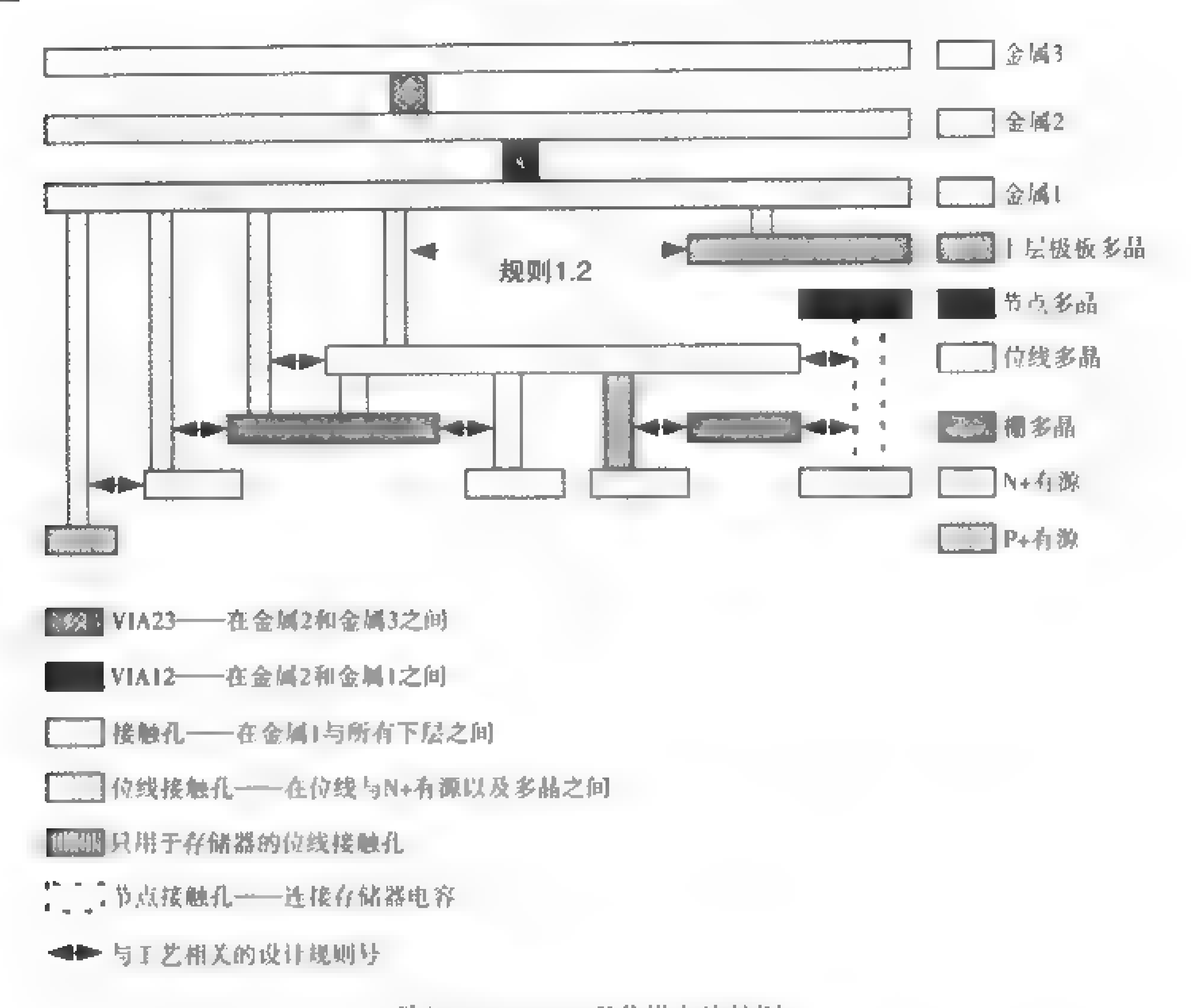


图3.20 DRAM工艺纵向连接图

3.6 通用设计步骤

图3.21给出了一种适用于所有版图设计类型的通用设计流程。

此流程简单易懂、不解自明，并且几乎可以应用在任何工程任务中。当然，这里我们将重点研究这一流程在版图设计中的应用。

我们会详细讨论步骤1，因为该步骤对于我们以正确方式开始设计版图是至关重要的。在这一步骤中，我们会收集并回顾所有相关的版图设计知识，并将其应用到被考虑的特定电路设计中。这一步骤的目的在于，尽可能对所有组元和信号所在的总区域进行文档记录，并在此基础上制定出整个版图设计的策略。

步骤2是设计的简单实现：基于设计的真实实现来进行版图规划，并且可能对它进行某些修改。可以将设计过程想像为这样一个过程：首先"自顶向下地规划"，然后"自底向上地实现"。也就是说，从顶层角度来看，我们首先会对总区域和通道进行版图规划，规划好顶层

布局后，我们从设计最底层的组元开始，并将其填入版图规划中所确定的区域，从而实现整个版图设计。实现较低层的设计任务会更加容易，这是因为在顶层版图规划中定义了施加给它们的约束。当完成总区域的规划后，我们就可以为下一步工作调整规划。如果版图规划很合理，那么只须做细微的调整。相应地，版图设计的完成就会变得非常容易。

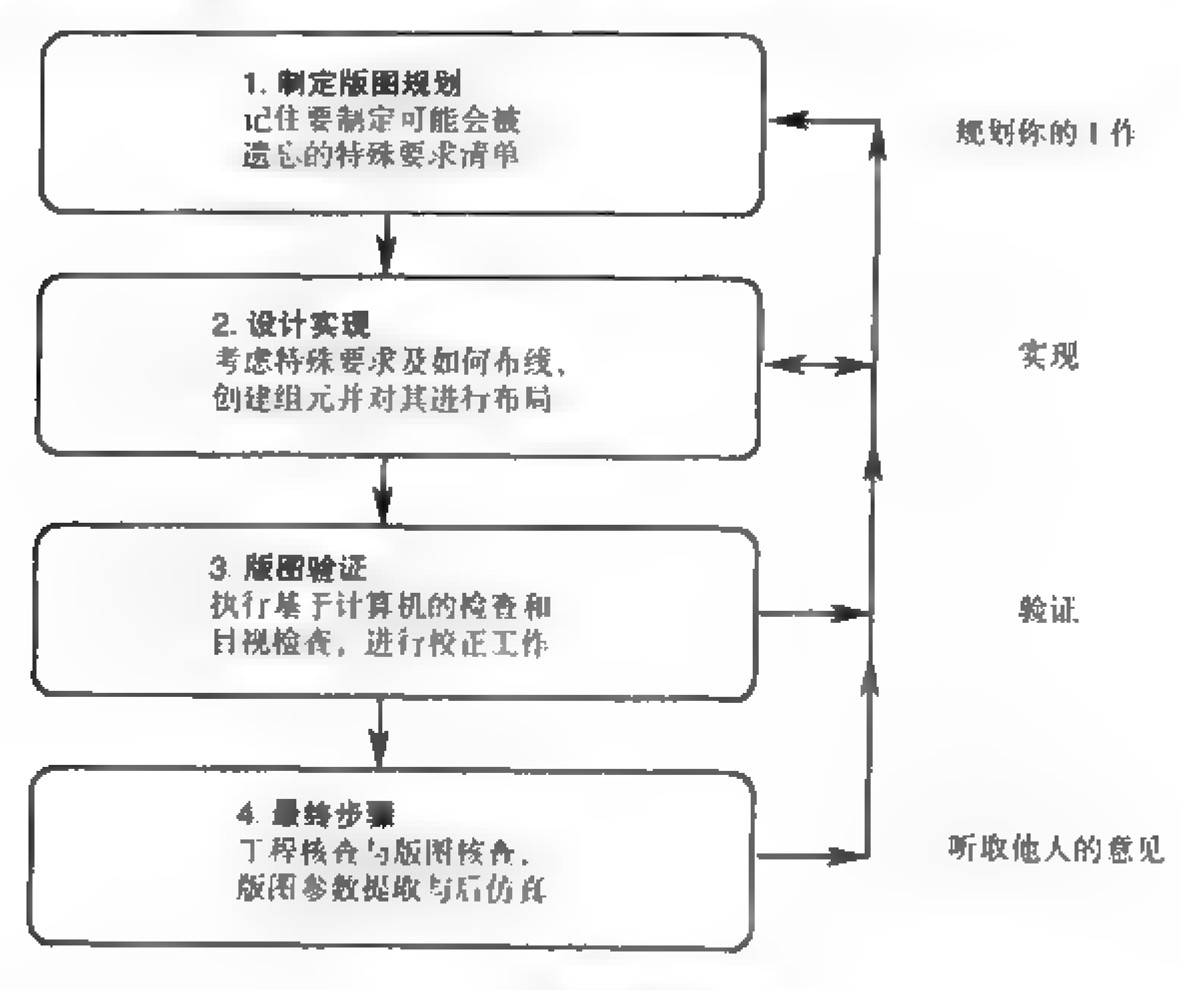


图3.21 版图设计步骤

步骤3的大部分工作是基于计算机的检查（将在3.10节中说明）。我们应该以3.10节中所述的固定顺序来完成这些检查。除了基于计算机的检查之外，建议还进行目视检查，这是因为计算机的自动检查实际上只依赖于所输入的规则。将设计打印出来并对它进行观察有时会更加可靠，另外，大多数设计的许多方面单靠计算机是检查不出来的，比如说，对称版图的对称度等。因此，应该将这些目视检查项列入核查清单以示提醒。

步骤4是最终的完整性检查和交叉检查，它和最终版图参数提取步骤一起，目的是确定是否所有的要求都已被满足且没有遗漏。

对于图3.21中所述的步骤和流程图，以及本章中介绍的所有步骤，有必要做一个特别说明：在设计有所修改和改进的每一步骤之后，都应该返回到前面的步骤中，去修改那些可能不再有效的要求。应注意的是，在该流程中是没有捷径的。图右侧的所有箭头只是向上的，因此每次修改之后，都需要重新执行前面的所有步骤。

3.7 准备开始

对于任何一种版图设计来说，最重要的阶段是规划阶段。在版图设计中，所谓高质量就意味着最后的结果即最终的版图能够符合用户（例如设计工程师）的要求。为了获得高质量的结果，版图设计工程师必须准备一份输入需求列表，将所有的规范和一系列输出要求都考虑进去，以确保满足输出版图要求。

3.7.1 制定版图规划

既然我们已经掌握了如何构建单一晶体管的版图，也理解了蕴涵在设计规则和制造工艺背后的相关概念，我们现在就可以利用这些基础知识来规划我们的版图。

我们必须记住在第2章中所叙述的一些概念。那些提供给我们来进行版图设计的电路图或者网表，常常会有许多不成文的或隐含的电学特性和性能特性，而这些特性恰恰是优化设计所需要实现的。理想情况是，除电路图之外，还应有一份成文的要求清单提供给版图设计工程师。请要求得到这样一份清单！

我们有一个建议，可能会在版图设计过程极其有益，但在所述的设计步骤中并没有指出来。那就是，在版图设计开始之前，最好花些时间来熟悉一下相关电路，这取决于你对所要设计的版图类型的熟悉程度。尝试一些从来没有做过的事情是非常少有的。

请参考过去的一些相同类型或相同工艺的设计。请咨询此类电路方面的专家。如果距你最后一次做此类版图设计已经有一段时间了，那么请复习一下本书中介绍的相关概念。在设计之前先花适量的时间去寻找一些在工作中可被重用并且可以帮助你的信息，这通常会给你带来一些好处。在开始工作之前，最好先听听别人的意见（通过借鉴别人的思想）！

当然，在另一方面，这一方法同样要求把你的工作经验和知识与其他人共享。

为了确保不会遗漏任何细节，你应该根据自己所做的研究，整理出一份版图设计前的清单，这对于制定版图设计策略是很有帮助的。图3.22给出了基于电路设计要求创建版图规划通用步骤的一个示例。

步骤1.1涉及到电源和/或全局信号的版图规划。电源连接通常称为电源网格。规划中必须考虑到从接口到该设计的各部分之间的电源电阻，特别是电源线的宽度和该设计的电源线网格。另外，同其他设计的接口也十分重要，尤其对于那些需要对一个单元进行排列的或者需要和其他单元进行无缝接合的单元设计。同时，不应忘记阱接触孔和衬底接触孔通常都是连接到电源上的，因此与其相关的版图设计策略也必须加以考虑。

步骤1.2是列出所有的输入信号和输出信号，并且在该设计与相邻设计之间的接口处为每个信号指定位置。接口被定义为设计的边界。在一些情况下，可能会为某些信号指定一个特定的或者非默认的信号宽度。需要进行特殊考虑的信号包括时钟信号、拥有多位（这些位之间需要相互匹配）的信号总线、关键路径信号以及屏蔽信号

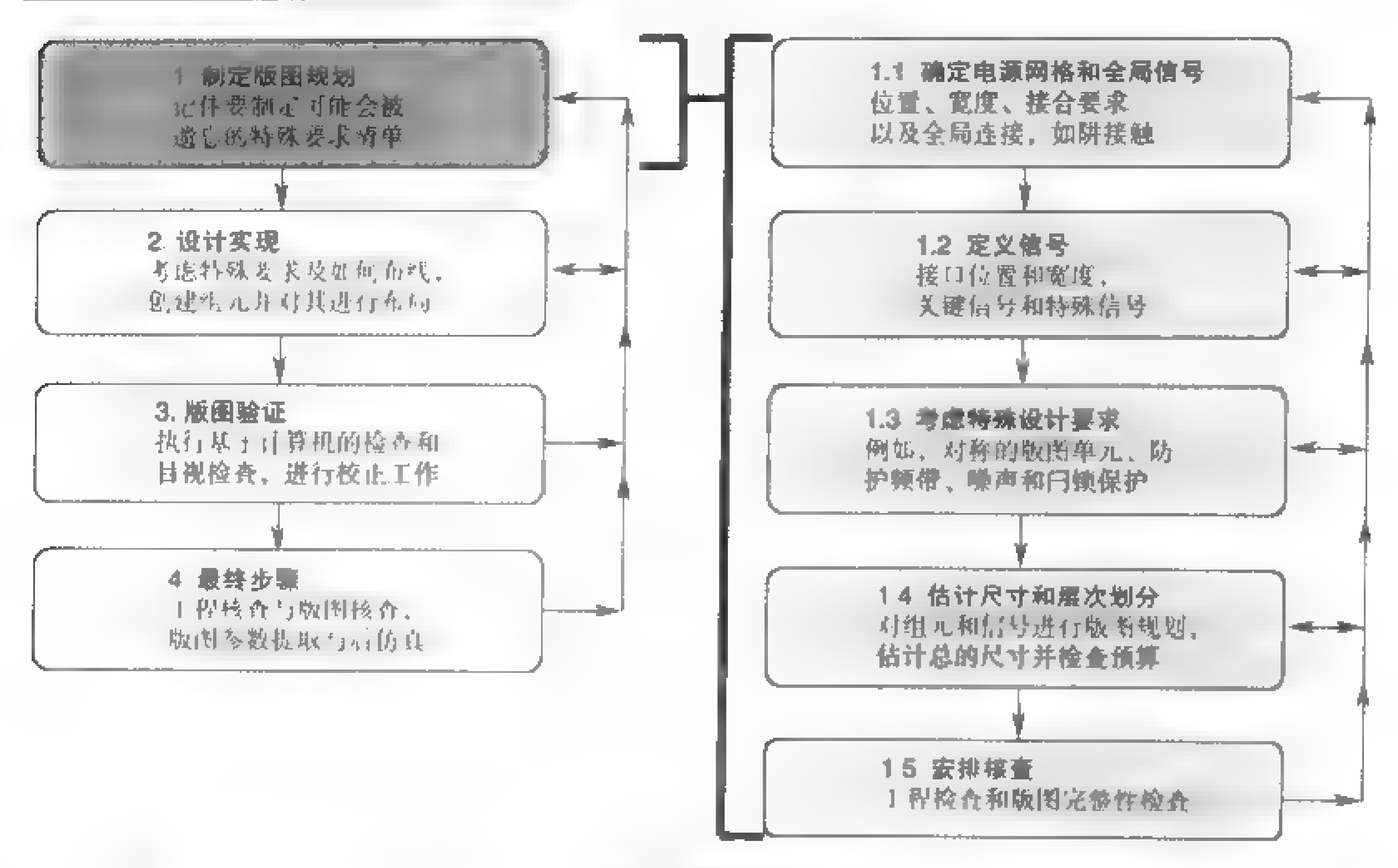


图3.22 版图规划步骤

在步骤1.3中，我们必须处理一些特殊设计要求，例如版图对称性、闩锁保护的特殊要求或者抗扰性。其他一些特殊设计要求的例子还有：设计必须是间距匹配的（即在某个方向上尺寸是受限的），设计必须有很明确的关键路径信号，或者可能是设计中的非标准部件。

步骤1.4非常重要，它有助于最终确定设计尺寸，以及估计在面积约束和进度限制下满足所有设计要求的可行性。依据从之前的具有相同复杂度和工艺设计规则的设计中获得的相关知识，版图设计工程师既可以估计出每个组元以及整个设计的尺寸，也可以确定需要实现的不同组元的数目，以及完成该设计的整体层次划分或区域划分。并且还应指定进行内部布线和信号连接的区域；确定每个互连区域中的布线层。至于最终设计的尺寸，由于我们仅能做一些有根据的估计，因此应该保留一些备用的信号和空间。

至此，我们应该有一个版图规划和版图实现的初步策略了。版图规划中应该先将所有信号端口指定到合适的位置上，然后在此基础上定义接口或边界。应该确定那些有特殊要求的信号，在总面积的估算中应包含这些特殊信号所带来的面积影响。另外，如果该设计是分层设计，那么子组元也可以通过各自的接口来确定。最后，规划中还应包含备用空间和备用信号线。

步骤1.5是做完整性检查和交叉检查，目的是确定所有的要求都被满足且没有遗漏。这些要求中，有些既和版图设计准则及工艺方式相关，又和电路设计要求相关。由于电路设计工程师很可能为自己的设计定义了一些特殊要求，并且当他们在设计环境中为设计建模时，会采用某种类型的版图规划，因此版图规划是版图设计工程师和电路设计工程师之间的交流工具，让电路设计工程师参与到版图规划的核查过程中是很重要的。

版图核查也涉及到你所设计的部分下一级的集成问题。正在进行规划的"块"必须与相邻"块"以及它们的接口完美地相接合，否则，当它们集成在一起时便无法按计划准确地工作了。例如，顶层芯片设计的尺寸必须在芯片封装的范围内，因此对所有版图规划进行检查是很重要的。负责集成设计的工程师应该核查版图规划，同时也要检查与设计尺寸相关的要求、版图结构或通道以及设计的接口。

需要两个人来核查那些确实复杂的版图规划并不罕见：一个人检查工程要求，另一个人检查版图要求。核查人员应该是没有直接参与设计的人员，应具有一定的专业经验，可以估计和评价版图规划质量，并给出建设性的意见。为了帮助核查人员进行正确的核查，对于各种版图设计类型，理论上应使用相应的核查清单。请参看附录中给出的核查清单。

3.7.2 棒形图

在准备版图设计时，棒形图（Stick Diagram）是对电路进行版图规划的一种简单方法。在很多情况下，绘制一些不考虑设计规则的简单版图草图，这对于电路设计工程师是非常有用的，可以使得他们能够更形象地想像出如何完成以及究竟是否能够完成版图设计。电路设计时往往会对最终版图形式做出某种假设，因此简单快捷的棒形图可以增加设计成功的机会。棒形图可以很简单，也可以很详尽，在此不做赘述。图3.23给出了一个棒形图的示例。

步骤1给出了所有器件的初步布局。在本例中，我们已经预先确定好电源线VDD和VSS的结构，同时也确定好了PMOS和NMOS晶体管的方向和位置。

步骤2给出了确定哪些有源区连接到相同电位上的过程，以及翻转器件（flipping device）的影响，以便于利用这些"共用"节点。

步骤3给出了共用有源的最终结果。事实上，这也就是互连最优的最终版图形式。这一步骤中的单元形状要比在前面步骤中的更窄。我们可以用棒形图的最终版本来作为提交核查的版图规划，具体内容可参见3.7.1节。

3.7.3 层次化设计

如3.7.1节所述，电路设计分析和最后得到的版图规划均说明了层次化设计的意义。层次化设计是指这样一种设计，它含有引用或使用其他组元作为自身结构的一部分，子组元又可以引用其他组元。这与计算机程序中子程序的概念类似。

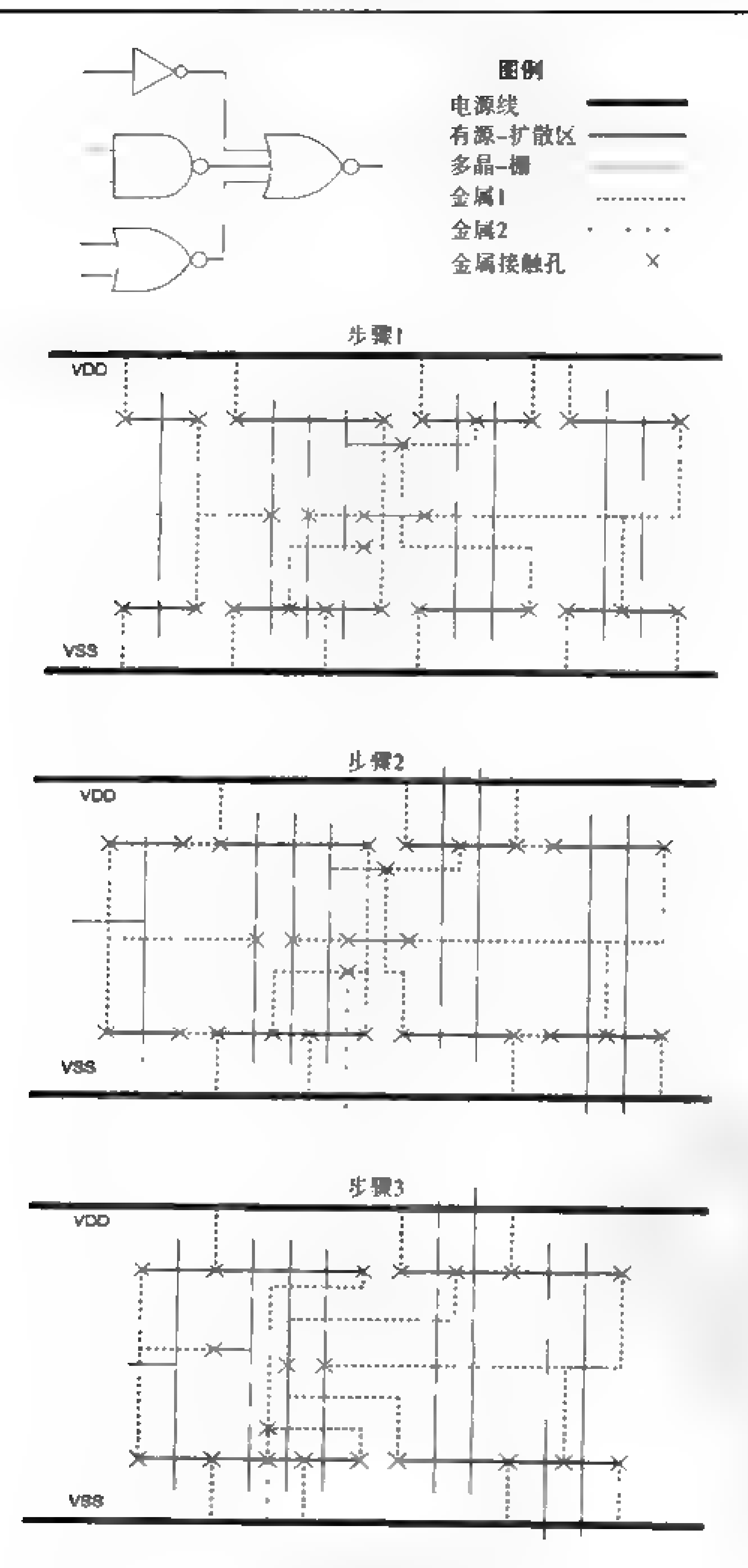


图3.23 棒形图版图示例

使用子组元构建设计非常有意义，原因如下：

- **计算机资源管理**：相比于进行单独的数据复制，已经存储在磁盘和存储空间的数据更容易被引用。
- **组元重用**：设计工程师可以重用那些已经被完成的组元——最好由该领域的专家对这些组元进行设计、验证及核查。
- **并行工程**：将一个设计划分为不同的子组元，将允许并行地完成多个不同的任务。

在版图设计中，我们可以将这些可重用的子组元称为叶单元。之所以称之为叶单元，是因为层次化设计就好像一棵树有一个树干、较多的枝干、更多的小枝干以及很多的叶子。

下面是一些关于叶单元的说明：

- 它们是可重用的版图设计，可以在芯片中的许多不同区域里重用。
- 它们可以仅由一个多边形构成，例如，接触孔单元；也可以由三个多边形组成，例如，带有环绕层（金属1、金属2、通孔12）的接触孔单元；或者可以是一个完整的电路，例如，反相器、与非门（NAND）、触发器。
- 对于同一个电路图，可能存在不同版本的版图设计，因为尺寸相同的反相器在芯片的I/O区中和在芯片的存储器区中其单元环境或者接口是不尽相同的。
- 任何使用标准接口的多边形组都可以作为叶单元。例如，几乎所有逻辑门的库都带有标准电源版图接口，因此每个逻辑门都可以设计成一个单元。如果要实现的是多重逻辑门的设计，那么不建议在晶体管级实现设计，而建议首先从制作逻辑门单元开始。
- 如果需要对设计进行全局修改，单元的使用将使这一工作变得更加容易。让我们来想像一下如果需要修改一个有100个反相器的设计。一种情况是，设计中使用了反相器单元，并将其例化了100次；另一种情况是，“展平式”设计中通过连接200个晶体管来实现100个反相器，而现在需要对所有100个反相器进行修改。在使用反相器单元的设计中，只需要修改反相器单元即可；根据使用200个晶体管的展平式设计中所需要做的修改来看，或许从头开始使用单元进行设计要比修改所有的200个晶体管的效率更高。
- 与上述情况相反，同样在这100个反相器的例子中，如果100个单元中仅有一个需要修改，那么我们就必须非常小心谨慎，因为即使只修改其中一个反相器单元，也必然会影响到另外99个反相器单元。在这种情况下，就需要制作能够反映所需修改的新的反相器单元，并且用其替代需要进行修改的反相器
- 设计中的每个单元都需要有一个惟一的标识符，即使该单元是对已经设计好的单元的再次例化，也应这样做。在上述的将一个称为INV单元例化100次的例子中，我们需要对每一个例化单元进行惟一标识（例如INV001和INV002），并且如果该单元在电路设计中存在电气对应关系，那么它的标志符应该和电路设计使用的名字相匹配。例化名用来

区分每一个实际上相同的INV单元。在我们所举的对100个反相器中的一个进行修改的例子中，例化名非常有用，因为我们可以用它来标识修改过的反相器。

- 单元比多边形组更易于翻转和旋转。
- 如果需要的是对称的版图，那么只需要用一个单元来描述半个设计，这一方法可以保证版图是完全对称的！
- 尽管由于多边形坐标偏离网格，造成了一定的风险，但单元仍可以被按比例缩放。
- 由于单元内部的所有多边形并不需要被显示出来，因此单元的使用还可以使计算机屏幕刷新资源最小化。通常只需显示一些关键的接口多边形，而将单元的实现细节隐藏起来。
- 可以将单元"排成阵列"，以便于节省更多的电脑资源。阵列可以被认为是一个单元矩阵的定义。比如说，要实现一个10 × 10存储单元矩阵，而我们有一个作为叶单元的单独存储单元。一种选择是将此存储单元例化100次（这会使得我们不仅要存储100个*X*、*Y*坐标，或者是200个数值，同样还需要能够生成这些数值），而更好的选择是定义一个阵列，它仅要求有*X*、*Y*的原点值、*X*、*Y*的偏移量，行数和列数（仅六个数值）。
- 单元阵列的使用还可以减小计算机屏幕刷新的时间。例如，某些软件包既可以仅选择显示一个阵列的边界单元，也可以选择显示边角单元。
- 相对于许多单独的单元，层次化版图的验证工具可以从较少单元的重复使用中得到好处。在上述对单一的反相器单元进行100次例化的例子中，版图验证工具只需对反相器单元进行一次验证，然后检查一下各单元之间的连接就可以了。相比于验证由200个晶体管连接成100个反相器的设计，这样的方式会使验证更加快捷。

图3.24给出了在一个设计中单元例化的例子，它包含例化名并使用了不同的方向。位于单元原点的具有固定格式的字母F向版图设计工程师显示了单元方向。

根据图3.24，有以下几点需要注意：

- 在所有情形中，基础单元名称为AGBC。
- PDC是顶层单元/模块的名称。
- 在设计中，每一次对AGBC进行例化，其例化名都是不同的，如comp001、comp002、comp003。
- 在第一个例子中，相对于PDC模块的原点，所有的单元方向为0°方向。
- 第二个例子中的例化单元comp002相对于其他例化单元是翻转的——注意两个箭头确定了镜像轴。
- 第三个例子给出了一个阵列实现。

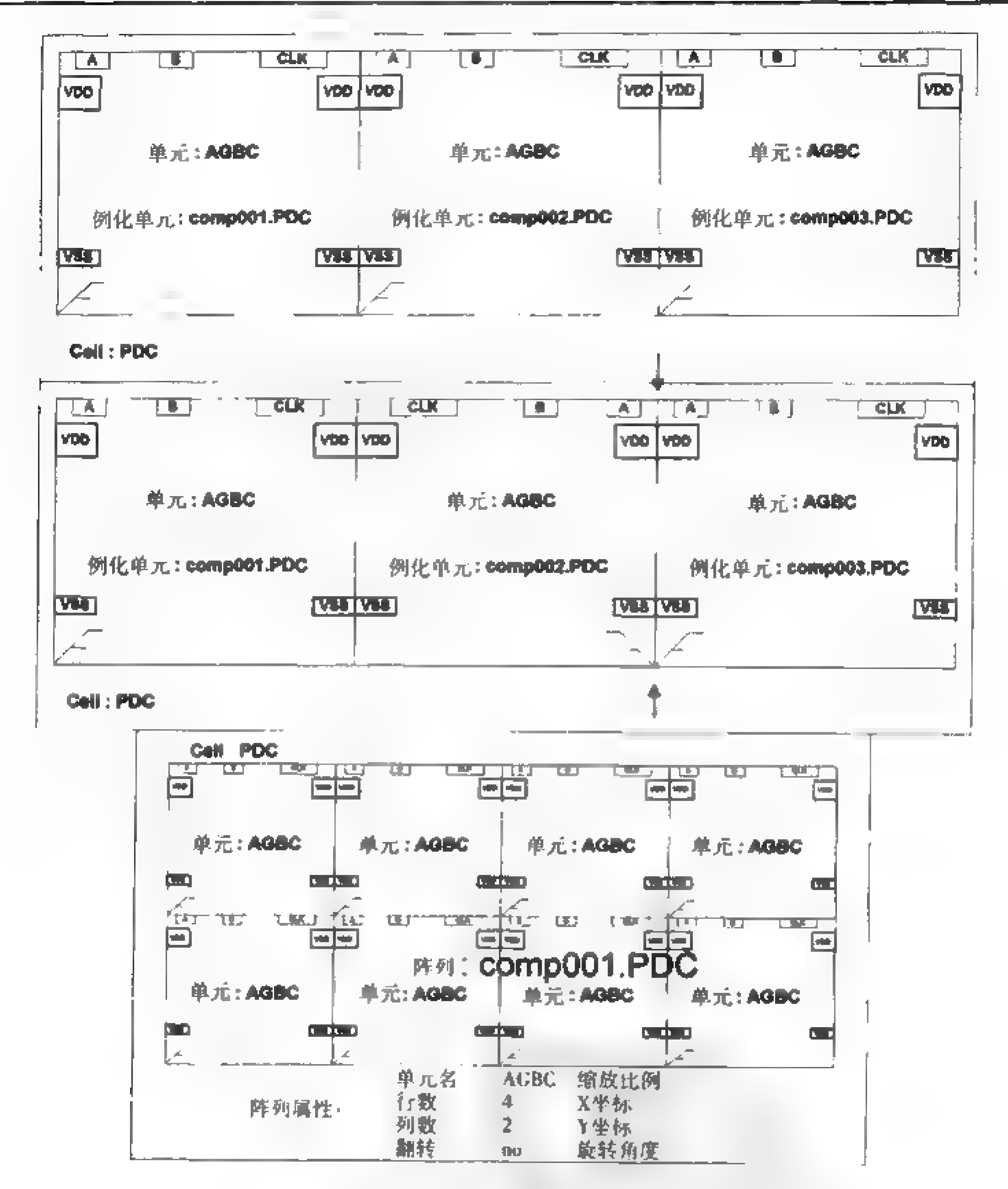


图3.24 层次化设计示例

3.8 通用准则

既然我们已经仔细考虑了版图规划，那么现在就应该开始实现版图设计。不论是在版图规划还是在版图设计的实现过程中，都有一些应该遵循的通用准则。本节列出了一些基本的准则。

3.8.1 电源线版图设计准则

在开始进行任一单元的版图设计之前，必须先确定电源线。电源线版图设计的特定准则如下：

- 基于下述准则来确定线宽：
  电源线是仅给单元内部供电，还是需要为其他单元供电而作为芯片电源网格中的一部分？这一关键信息可以根据版图规划确定下来。
  利用不同分层的电阻率来确定合适的线宽。
- 使用最底层金属作为晶体管级单元的电源线。因为必须考虑到，如果使用高层金属作为电源线，那么就需要通过通孔和局部互连多边形来连接晶体管和电源，这样会占用空间并且限制单元的空隙率。因此，通常会使用工艺和设计所允许的最底层金属作为电源线。
- 避免在电源线上开槽。电源线上会通过大量的电流；因此，重要的是应确保以一致的线宽对电源线进行布线，并且不在线上开槽。线上的任一开槽都可能使该处的电源线像熔丝一样，在强电流情况下电源线可能会断裂。
- 避免在单元上方布电源线。除非使用自动布线工具，否则不推荐在单元上方布电源线。将电源线布在单元内部来确保单元可以“一次性构建即可保证正确无误”。若将电源线布在单元外部，理论上则限制了单元之间的互连。

3.8.2 信号线版图设计准则

下面给出专门针对信号线布线的准则。在后续章节中还将对这些准则进行详细的说明。

- 基于工艺参数和电路要求选择布线层。对于每一种工艺，应该根据分层的电阻和电容参数来确定所有的标准布线层，而诸如N阱、有源、高阻多晶栅等分层则不能用于布线。使用相同的标准还可以使布线层间的优先级标准化。
- 使输入信号线宽度最小化。使设计的信号线布线最小化也很重要，这样可以降低信号线的输入电容。当信号作为单元的一部分，需被多次使用时，这一点尤其重要，例如一个单元内的时钟信号线。
- 谨慎地选择布线宽度。选择信号线的布线宽度必须深思熟虑。将最小的设计规则线宽作为实际布线宽度是很吸引人的。但是，考虑到必须在每条线之间进行连接，这一方式并不总是切实可行的。

如在图3.25中看到的那样，这些连接点需要通孔或接触孔来进行连接，而通孔或接触孔所需的空间通常比布线层所要求的最小宽度规则大。

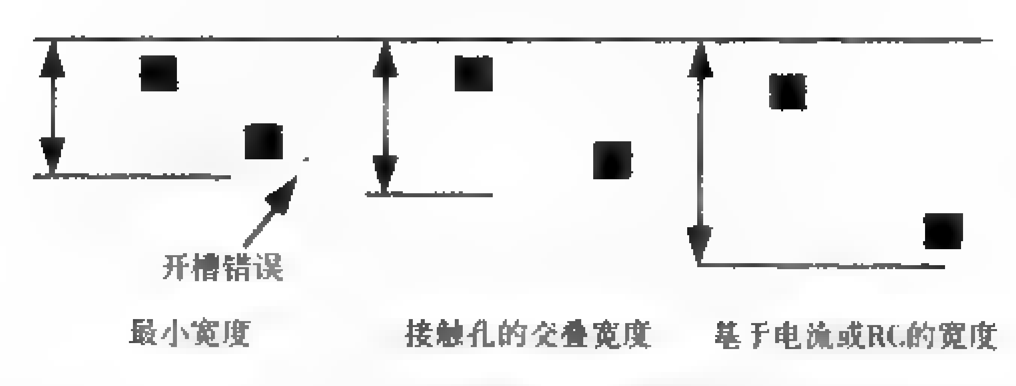


图3.25　对布线宽度的考虑

使用既能容纳接触孔又能容纳通孔的线宽可能会使设计更加有效。这种线的线宽虽然会宽一些，但是由于它避免了改变接触点周围的其他信号线，因此更易于处理和维护。

- 在同一单元或模块中保持一致的布线方向（见图3.26）。这样的话，当版图设计工程师想改变信号的方向时，就只需使用一种通孔类型，且不会影响单元的空隙率。通常，对于每个分层来说，保持一致的金属方向，并且和相邻分层的金属方向交错开来是很有意义的。例如，如果金属1、金属3、金属5水平地布线，则金属2、金属4、金属6应该垂直地布线。

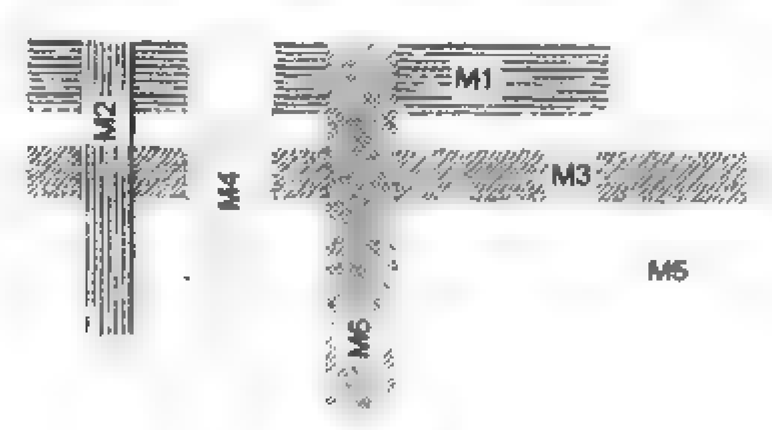


图3.26　布线方向

- 标注出所有重要信号。这对版图验证过程尤其是LVS是非常重要的。当节点被标注出来后，错误诊断、短路排查都将变得容易，LVS运行时间也会缩短。
- 确定每个连接的最小接触孔数。不要认为每个连接上仅使用单个接触孔或通孔就已足够。比方说，某些存储器会尽可能地使用双接触孔来增加可靠性。

3.8.3　晶体管版图设计准则

下面给出专门针对单元级设计环境的设计准则。

- 使用预先定义的模板来进行PMOS和NMOS晶体管的布局。应预先定义好单元的结构，并且该模板应将一组单元的基本版图规划封装起来。图3.27给出了一个单元模板的示例。
- 使用指状晶体管来实现大的晶体管和关键的晶体管。与图3.27所示相似的单元模板根据单元高度定义了晶体管的最大宽度，而若要设计一个晶体管版图，其宽度超过了此高度，那又该如何呢？办法是将晶体管分成像“手指状”的多个并联晶体管。

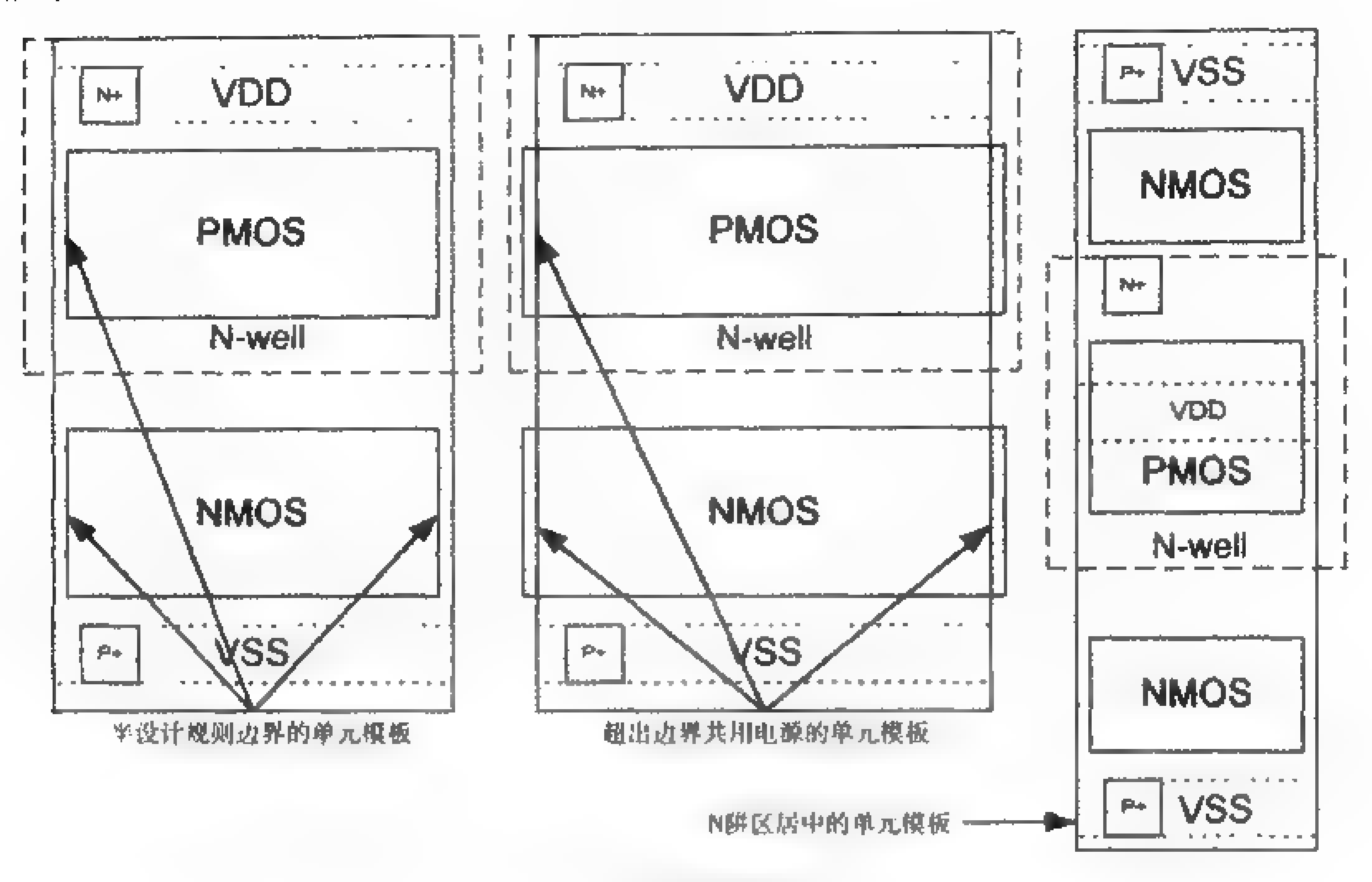


图3.27 单元模板示例

图3.28给出了100 μm宽晶体管的三种等效的版图设计。

偶数个指状晶体管有这样的优势：有效电容较低，这是因为包围晶体管漏区的是多晶硅栅而不是场区（field）。

使用指状晶体管的另一个原因是为了优化由晶体管宽度所引起的多晶硅栅电阻。因为多晶栅是由单端驱动的，并且是有电阻的，所以需要一个准则来规定单个指状晶体管的最大宽度。因此，对于大晶体管来说，将其设计成多个指状晶体管是遵守最大宽度准则的惟一方法。

将多个串联晶体管设计成指状是需要技巧的。图3.29给出了一个将两输入与非门（NAND）设计成指状构造的示例。

将PMOS器件设计成指状相对简单；但是，将串联的NMOS器件设计成指状会比较困难，因为我们必须保持这些器件的连接顺序不变。

- 共用电源节点以节省面积。我们很容易理解尽可能地共用节点这一设计理念。由于电源节点分布广泛、易于连接，因此实现共享相当容易。电源共享可以节省相当大的面积。共用电源节点能够节省面积的主要原因是，一排接触孔的两侧都会被使用到，因此没有必要将两个有源区相互分隔开来。图3.30说明了这一方法。

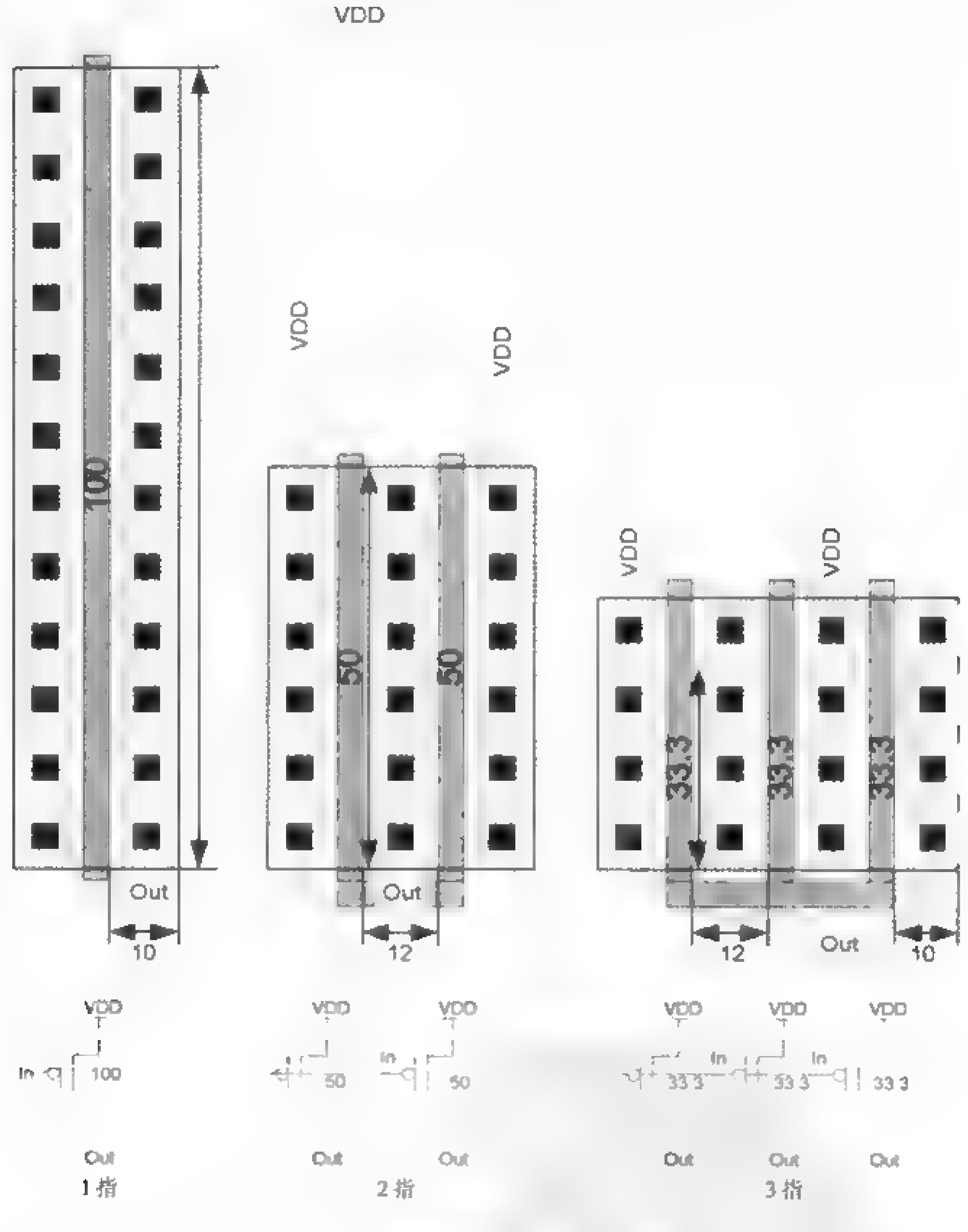


图 3.28 将晶体管设计成指状构造

要实现不同宽度晶体管间的电源节点共享，需要在较小晶体管的多晶硅栅末端与有源之间留出一定间距。

- 确定源极连接和漏极连接所需接触孔的最小数目。关于这点，有一个简单规则可能适合，那就是在两个接触孔间尽可能多地使用最小设计规则。遵守这一准则的优点是非常可靠并且可以使晶体管性能最优，而缺点则是穿越晶体管的可布线性受到限制

如果需要提高可布线性，并已在电路设计中加以说明，那么对于整个版图设计来说，接触孔数目比最大值略少一些是最佳选择。使用这种方法必须经过仔细地考虑，并且必须在电路设计过程中加以说明。

通常情况下，ASIC单元库会在单元内使用最小数目的接触孔，但是对于某些高频设计或者模拟部件而言，晶体管需充分地接触

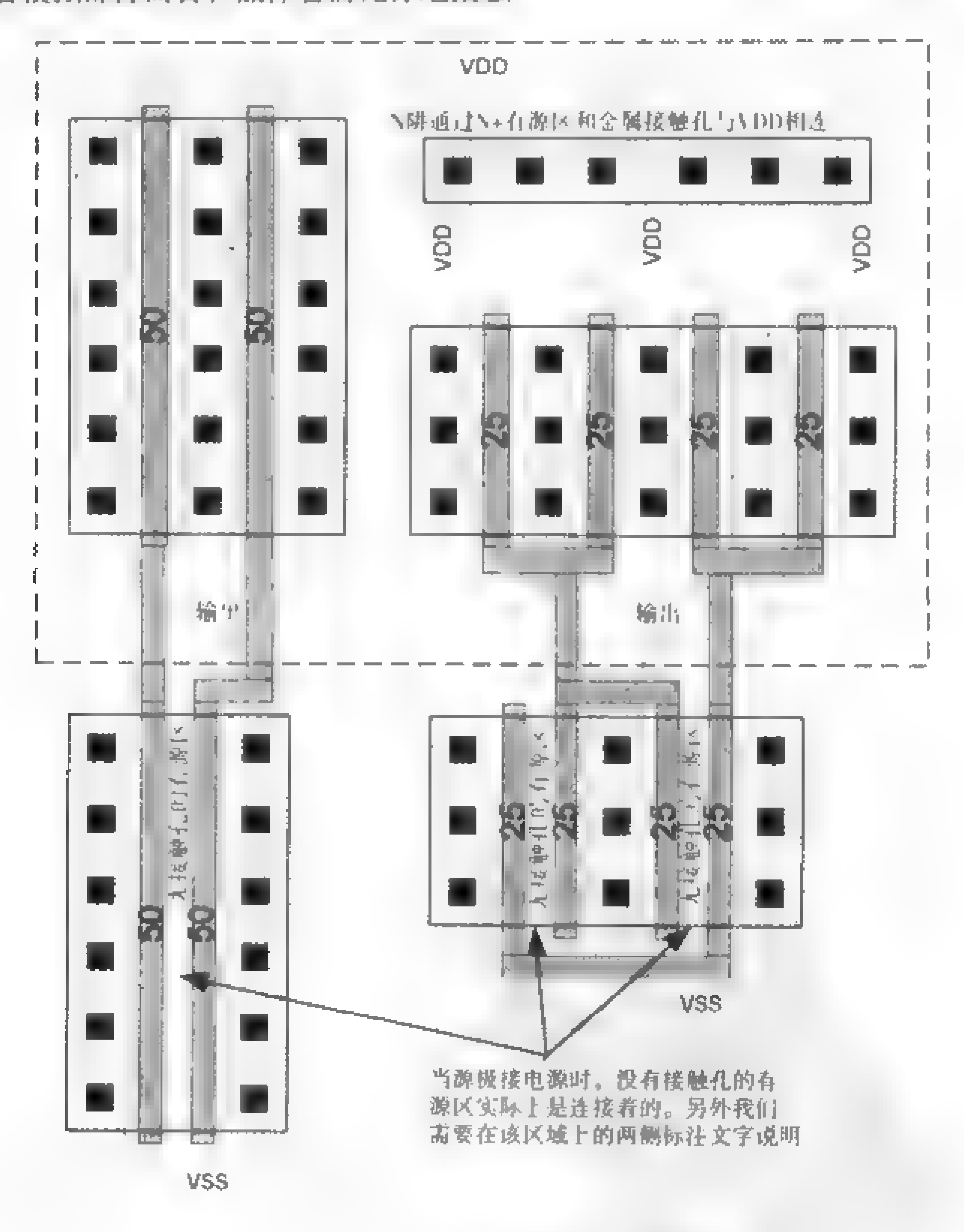


图3.29 将与非门设计成指状构造示例

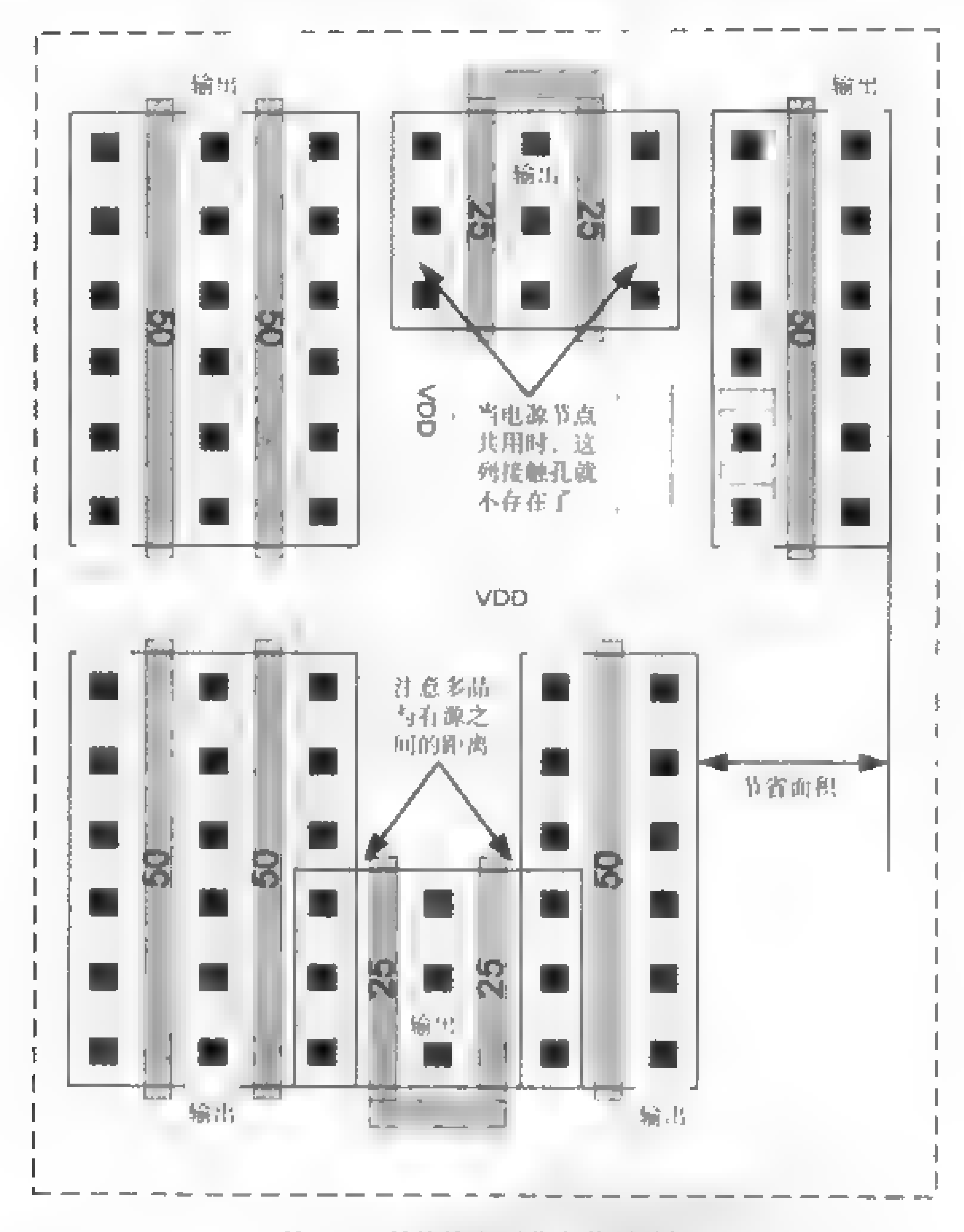


图3.30 晶体管电源节点共用示例

- 尽可能使用90°角的多边形或线形 大多数设计都采用这种方式，这是因为若用直角形状，计算机所需存储的数据量最小，版图设计过程也更易于实现

  对于有严格的面积和性能约束的区域，应限制45°角版图设计的使用，这是因为这种设计的修改和维护相对困难 对于一般的版图单元来说，花费额外的精力来使用这种45°角版图设计技术是不值得的

  而对于存储器单元和间距受限的版图，却值得使用45°角的版图设计，同样的情况还有数据通道和大功率电源线 但是，对于许多CAD程序来说，45°角的接触孔和通孔会带来很多问题，因此还是要避免使用它们。

- 对阱和衬底的连接位置进行规划并使其标准化。N阱与逻辑“1”电源（VDD、VCC、VPP）相连接，而P+衬底接地或VSS。

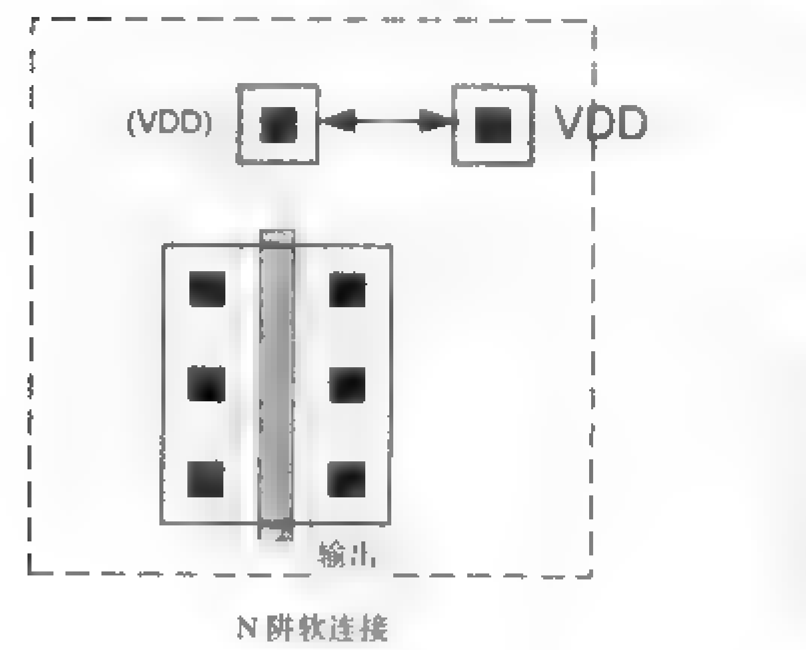
  单元的版图规划应包括放置这些接触孔的大体区域。有两种基本方法：将它们放置在PMOS晶体管和NMOS晶体管之间，或者将它们放置在晶体管区域之外。前一种方法更有利于闩锁效应保护（稍后进行讨论），但会增加单元版图设计的复杂度。

- 避免“软连接”节点。“软连接”节点是指通过非布线层进行连接的节点（见图3.31）。由于非布线层具有很高的阻抗，若通过它们进行连接，会导致电路性能变差，因此它们通常会被标注出来。

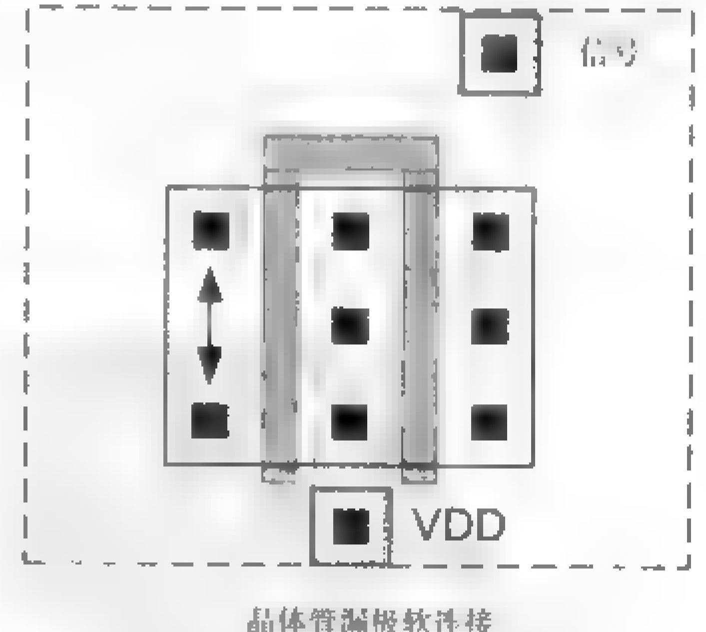
  有源层和N阱层通常不是布线层，但是仍有可能因为疏忽而用这些层来进行电气连接。这样一来，尽管DRC和LVS仍可以通过，但是整个电路的性能却会很差。只有进行非常细致的版图参数提取和后仿真才会发现这种“软”错误，但这样做通常不切实际，因此往往采用“一次性构建即可保证正确无误”的方法来避免这种错误。

  图3.31给出了两个软连接实例。N阱的例子显示了晶体管如何与VDD进行电气连接，但在这个例子中，信号线穿过N阱而成为连接的一部分。第二个例子说明了由于没有用金属来完成和晶体管漏区的连接，其性能可能会受到损害。在这个例子中，这个单独的接触孔无法起作用，因此在等效的版图中将不存在这个接触孔。

  在版图验证过程中加入特定的检查可以帮助发现这些问题。但是，调试更正这些错误是很困难的，因此从一开始就完全避免犯这些错误才是最好的方法。



N阱软连接



晶体管漏极软连接

图3.31 软连接示例

3.8.4 层次化版图设计准则

- 开发和使用版图规划。这一点是怎么强调也不为过，并且在版图设计各层次上都应如此。

- 在规划阶段确定设计的层次划分。对设计的层次划分并没有严格的规定，但一些常识性规则却也难以违背，确定不同层的层次划分的常用准则如下：
  有必要将被例化多次的电路模块定为单元。
  将设计分成功能模块或区域指定模块。
  将设计划分成一些允许多个设计工程师并行设计的模块。
  如果要求的是对称的版图设计，那么将单个半单元和其镜像组合在一起就可以完成对称的设计。
  本书的许多章节中都会讨论层次划分的使用。
- 提出一套针对靠近单元边界区域的版图设计标准，并在设计过程中遵守这一标准。在版图规划和正在实现的单元类型中，应该定义单元如何与其相邻单元相接合。在规划阶段就应熟悉并掌握所有的设计接口要求。
  这里给出一些如何正确设计单元接口的技巧和准则：
  使用模板单元来定义设计的全局特性——可以通过单元尺寸以及电源和阱的布局来确定单元的接口。坚持使用模板可以确保单元遵守设计标准，单元集成也因此得以顺利进行。
  假定边界接口是固定的——如果任一多边形需要穿过单元边界，那么在允许其通过之前，都需要参考设计中更高层次的版图规划，这样可以避免预料之外的多边形交叠。
  设计规则取半方法——如果单元将和自身相接合，或者与其他具有相似边界条件的单元相接合，那么“一次性构建即可保证正确无误”方法将确保单元内部所有多边形都距离单元边界一定值，而这个值是特定设计规则的一半。以这种方法，当该单元与另一单元相接合时，就不会违反间距规则。
  将单元和其相邻单元一起进行验证——这种方法可以确保单元在任何情况下都正确。

3.8.5 质量度量标准

一旦满足了基本要求，并且/或者我们拥有了一定的版图设计经验，就要考虑以下列表，它列出的议题和度量标准更加微妙，对此我们还将在后续章节中做进一步的讨论。在这里我们只将它们提出来进行简单的介绍，以便于我们能够对这些更高要求做事先的考虑以及大致的规划：

- 面积
- 性能
- 空隙率
- 可制造性（即设计尺寸是否全部为最小设计规则？如果不是的话，则认为其具有更高的可制造性）

- 可维护性（即工艺改变时，版图是否易于更改或优化？）
- 长期的可靠性（即电迁移）
- 接口兼容性（即是否适合于所有的例化情形？）
- 可收缩性（即是否即使将来工艺尺寸缩小，版图亦同样适用？）
- 可重用性（即版图是否适合于移植或是重定义到不同的工艺上？）
- 版图设计流程的兼容性（即版图是否和所有的后续工具和方法兼容，例如布局布线工具）

3.9 设计的实现

现在让我们将到目前为止所学的所有知识运用于设计的实现工作中。此时，我们已经拥有了完成一个基本版图设计的全部基础知识

首先，让我们来回顾一下前面阐述过的一些关键概念（见表3.1）。如果已掌握这些基础概念，那么就可以应付大多数的版图设计任务了。

表3.1 概念回顾

章节号	概念摘要	说明
1	什么是版图设计	也包括版图设计在IC设计流程中所处的位置
2	怎样分析电路图	电路图中所包含的信息比呈现在眼前的要多得多
3.2	分层的定义	CMOS工艺的简介
3.3	CMOS晶体管版图	基本版图介绍
3.4	设计规则	这些设计规则定义了你所设计的版图的限制
3.5	分层的连接	给出了连接的规定
3.6	设计步骤	通用操作指南
3.7	制定版图规划	可能是获得成功的最重要的步骤
3.8	通用准则	需要遵循的设计理念，以确保首次设计是正确的

我们已经知道，在开始任何版图设计之前，必须先制定一个计划。而在我们开始执行这一计划之前，非常重要的一点是要预先考虑下一步的工作是什么，并在实现设计时将其牢记在心。这样就可以减少或消除由自身疏忽而造成的重复工作次数。

基于版图规划的组元布局（图3.32中的步骤2.1）是另一个领域，在该领域中，我们为实现合适的组元布局所投入的精力，总会获得很好的回报。整个设计能否完成布线受限于版图组元的布局，这些组元可以是晶体管、阱接触孔、电源线或信号线接口位置，也可以是例化的组元。因此从长远来看，在布局阶段所花费的额外精力，会使我们获得一个更加易于布线的设计。

在这一阶段中，应对包括电源、信号、直通以及其他多边形在内的信号线或接口进行标识确定，以避免出现连接错误。

我们可以把步骤2.1看成是为了验证版图规划是否可行的一次初步布局，而在步骤2.2中，我们开始考虑更精确的细节，此时，我们必须处理一些特殊的设计要求，比如关键路径信号、

衬底接触、版图对称性、闩锁效应保护或者抗扰性的特殊要求等；在此步骤中，我们会进行详细的布局，也会进行重要信号和复杂信号的布线，如电源和时钟信号就属于这类信号。此外，在版图设计开始或完成之后，应考虑到可能新增的设计要求，为组元和布线分配备用空间。

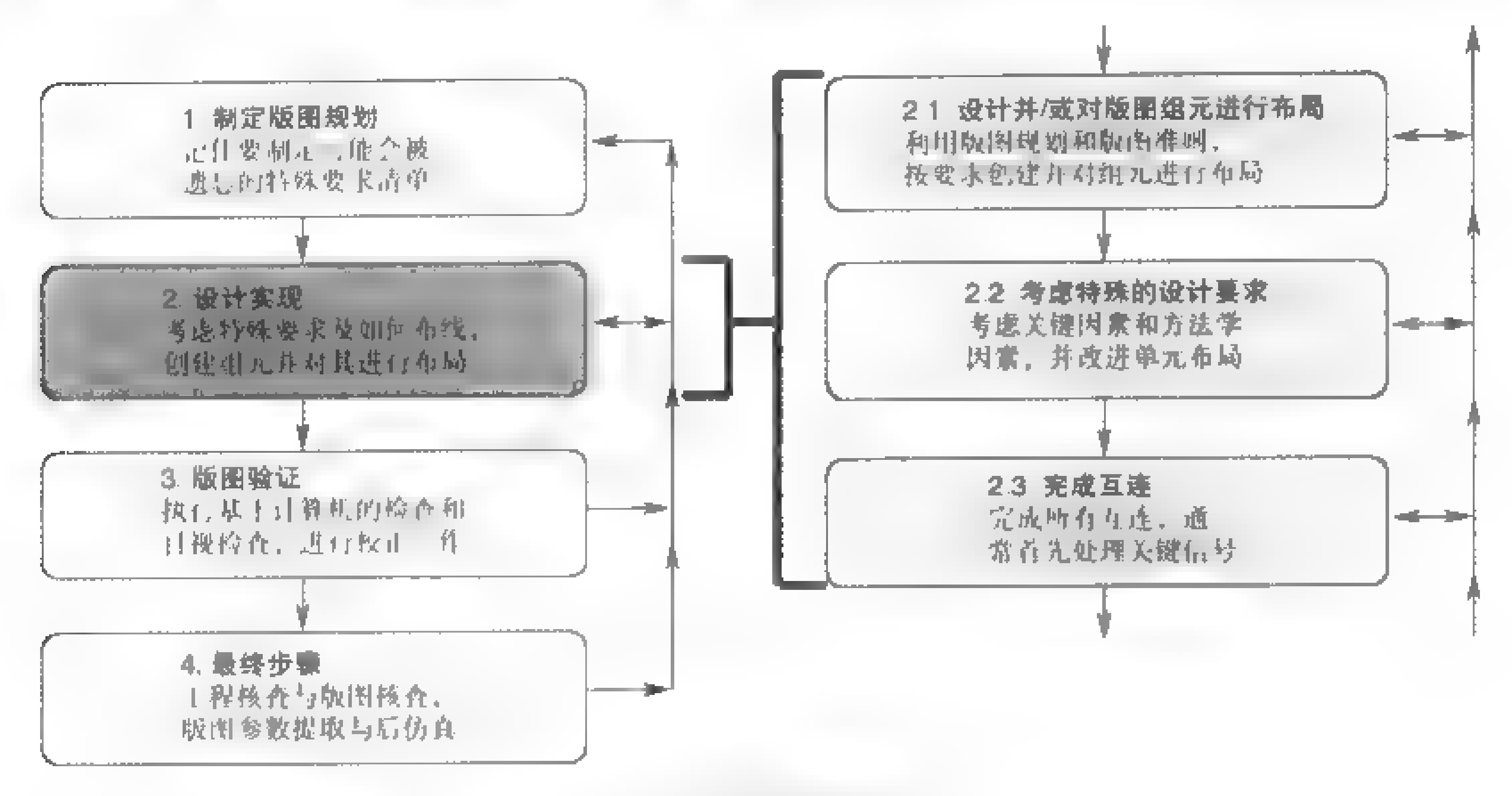


图 3.32 版图实现步骤

如果版图规划和版图组元的最终布局都完成得很好，那么步骤2.3将变得非常容易。而如果没有前面的步骤，那么当涉及到特殊设计要求时，完成互连布线通常会非常困难而且相当耗时。应当记住，在版图规划中应该仔细考虑布线层、布线方向以及所有信号的布线间距等问题，这样的话，完成布线就会变得非常简单。

现在让我们来分析一些与特定类型的版图设计相关的问题。

3.9.1 单元版图设计

通常，叶单元和逻辑单元是依据基于晶体管的电路图绘制的一种版图；因此这类设计中所使用的组元大多数会是多边形、晶体管、接触孔和信号引脚。由于我们是在最低抽象层次实现版图，并且我们需要完全了解整个分层集合以及版图设计规则，所以在此通常使用"Polygon pushing"的版图设计方式，而正式文献中将它称为"全定制"设计方式。

在这一级的版图设计上，我们提出以下几个关键的设计理念：

- 要完全了解整个分层集合以及版图设计规则。
- 根据设计中晶体管的数目及版图设计规则来估计设计的尺寸。

- 请注意为了实现逻辑门而进行的晶体管级布局和互连。
- 进行精心的版图规划和体系结构定义，以实现面积最小化而性能最优化。这些叶单元可能会被使用数千次，因此，花费额外的精力来尽量节省每一个单元的面积，而这在完成整个芯片设计后会得到应有的回报。
- 对电源实现进行精心的设计，这里也包括对衬底接触和阱接触的考虑。如果这一工作完成良好，那么要满足整个模块或是芯片的电源布线和基底连接要求，只需通过满足这些要求的叶单元来实现就可以了。
- 请注意与其他单元相接的接口设计。如前所述，叶单元可能会被使用许多次，因此，使两个叶单元相接合所需消耗的面积最小化可以节省版图面积。理想情况是将叶单元设计成具有与所有可能相邻的单元直接相接合的特性。

3.9.2 模块版图设计

说到单元与模块的区别，往往仁者见仁、智者见智，但一般说来，如果一个单元包含一组具有中等复杂度和功能的电路，并且主要由例化单元组成，那么它就称为模块。与单元相比，模块更大、更复杂。设计模块的工程师与设计单元的工程师相比，需要具有更加丰富的经验。由于这一设计方式将基于单元的设计与全定制设计结合起来，因此我们将它称为“半定制”设计方式，而图3.32中所描述的版图设计流程仍然适用。

对于模块级设计，需要牢记以下几个要素：

- 根据设计中单元的数目以及在单元内部和外部布线的信号数，来估计设计的尺寸。
- 模块设计中常见的现象是：分配大量的空间给备用组元和信号，以及一些可能仅穿过模块的信号，这些信号称为直通信号。
- 某些模块可能会包含功能不同的组元，比如数字单元与模拟单元的混合设计。在这种情况下，可能要特别考虑一下不同电路的要求，比如闩锁效应和抗扰性。

通常，当版图设计工程师获得了上述所有信息后，就可以开始对已有单元、晶体管（若有的话）以及互连布线进行布局了。

我们通常依据电路功能特性和版图设计类型来定义模块。常见的模块有行单元、标准单元、数据通道或寄存器组、I/O（输入输出）环、模拟模块和存储器模块。

3.9.3 芯片版图设计

实现整个芯片的版图设计与实现大的模块的版图设计，两者从版图规划直到版图核查的所有步骤都是相同的，因此两者的设计理念也是相同的。

整个芯片的版图设计包含不同类型的模块和单元，所以需要详细了解所有版图设计方式，以使成功的概率最大。

以下几点通常只出现在芯片级的版图设计中，必须给予充分注意：

- 大型模块的设计划分。对复杂芯片中的众多接口进行定义是一项复杂的工作。
- 对芯片与外界的接口进行定义和规划。这里涉及到Pad、I/O电路以及与它们相关联的复杂电路的知识。
- 规划并实现那些遍及整个芯片的关键信号。其中包括全局电源线和时钟信号。
- 版图规划的技巧和维护在这里是最重要的。在整个芯片级版图设计中，版图规划作为交流工具和版图实现工具一样重要。
- 估计芯片尺寸本身是一项具有重要意义的工作。过去的设计及经验对估计芯片尺寸很有帮助。同时，应将当前项目中的工艺参数与先前项目的工艺参数进行比较，而这正是版图规划工具能够真正发挥作用的领域。
- 作为版图设计项目负责人，需要对整个芯片版图的设计负责，还要负责确定版图设计方法、进行任务分配，以及制定整个团队的工作计划。熟悉版图设计工具及验证工具可以确保团队高效地工作。

总之，正是整体芯片版图设计任务的复杂度，使其成为了版图设计中最具挑战性和趣味性的工作之一。

3.10 验证

由于已经完成版图的实现工作，接下来，我们就进入到验证这一步骤了（见图3.33）。验证工作并非微不足道，相反，它相当重要。虽然IC设计中存在许多故障机制，但修复错误的代价却非常昂贵。与修理汽车不同的是，修理汽车相对便捷，只需找到损坏元件来做更换或修理即可，而修复IC设计错误则需要花费数月的时间。正因为这个过程漫长而昂贵（这与试图修理已经在轨道上运行的卫星有些相似——有可能实现，但是耗费非常大），我们应该争取一次成功。

为了得到最优的结果，除了所有的规划和核查清单外，还需要一份详细的验证计划。这一计划中的每个步骤都会对设计中的不同方面进行检查。

3.10.1 设计规则检查

设计规则验证步骤会检查版图数据中的所有多边形和分层是否遵守制造工艺规则。如3.4节中所述，这些设计规则定义了可制造设计的尺寸限制。宽度规则和间距规则就属于这一范畴。

满足制造要求就是必须检查和校正的最基本的规则集合。因为设计规则检查（DRC）是版图实现后所进行的第一级验证，所以通常也会对相应的方法、连接关系以及指导性规则进行检

查，我们将之称为补充规则　比如说，分层的非法使用（在逻辑区域使用ESD层）、非法的器件或连接都属于这些规则范围。

提示：真正完整的DRC验证步骤并不局限于对版图设计工程师所实现的版图设计进行单独验证，还会将设计放置于应用环境中对设计进行验证

如果与该设计相接合或者相邻接的特定组元是可用的，那么将该接口单元和设计一起进行DRC检查　如果该单元是一个通用设计，那么应该进行更加复杂和彻底的检查，这包括所有可能的接口单元，以及可能出现的不同方向和组合情况　在将设计集成到整个芯片时，这些验证步骤确实可以消除出现错误的可能性

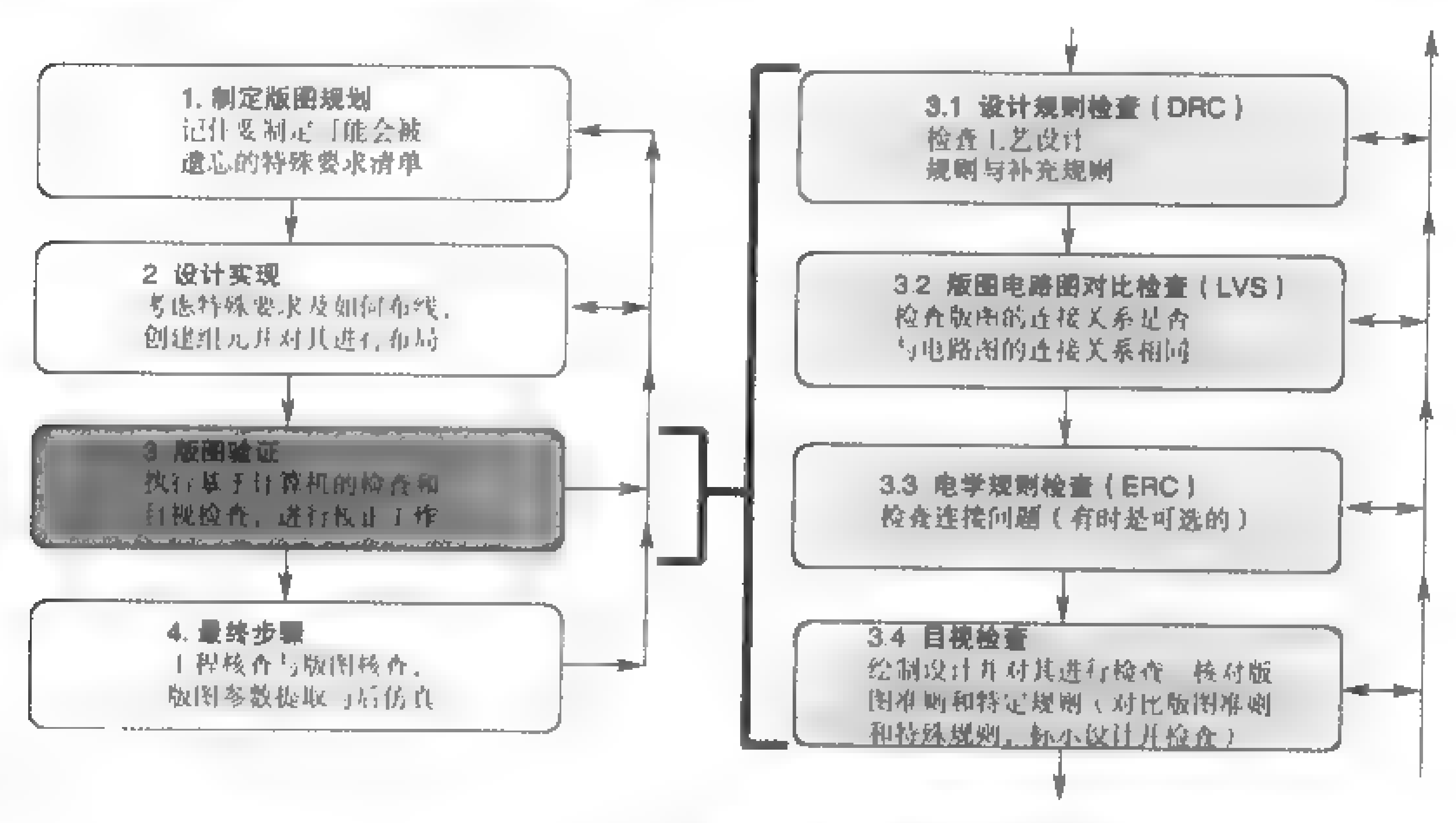


图3.33　版图验证步骤

3.10.2　版图电路图对比检查

版本电路图对比检查（LVS）验证用于检查设计是否正确连接　电路图是参考电路，版图应和它进行核对。原则上应对以下几方面进行验证：

- 所有信号的电气连接关系　包括输入、输出，以及电源信号与相应器件的连接
- 器件尺寸：晶体管的宽度和长度，电阻大小，电容大小
- 识别未包括在电路图中的备用组元和信号，悬空节点就是一个实例

最后一条与下面所描述的电学规则检查中的一条重叠

3.10.3　电学规则检查

如图3.33所示，电学规则检查（ERC）有时是可选的，或者很少作为一个单独的验证步骤来使用。因为许多问题用LVS检查就可以发现，因此ERC显得有点多余。

在这一步，通常只有与连接关系或器件连接错误相关的电学规则需要进行ERC检查，比如：

- 未连接、部分连接或备用器件
- 无效晶体管
- 悬空节点
- 短路
- 在其他步骤中不进行检查的特殊检查（例如天线规则）

作为LVS检查的子集，ERC检查通常执行得更快，因此它对于加快调试问题，如电源地（VDD-to-VSS）的短路错误很有帮助。

3.11　最终步骤

尽管IC设计工具在过去的10年里取得了极大的进步，并且顶尖的设计工程师拥有最佳设计方案，但是总有改进的空间，总有忽略某些事情的可能性。随着时间的推移，工艺设计规则的复杂性和多样性也会给旧的问题带来新的挑战。因此，以核查为形式的最终完整性检查和交叉检查是值得花时间的（见图3.34）。

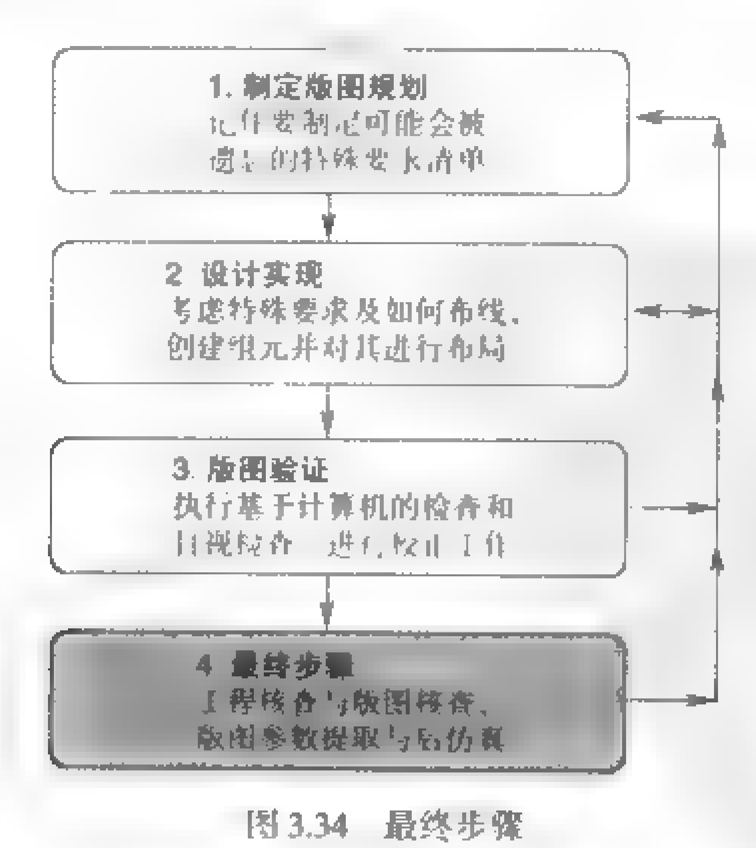


图3.34　最终步骤

相比于很多年前，现在的核查清单规模可能减小了，但核查的负担并未减轻，并已完全转移到了负责安装和维护版图CAD工具和版图设计流程的专业人员身上。CAD验证工具可以检查很多问题，但是仍需要有经验的人去确保工具正常工作。

获得正确核查结果的秘诀在于拥有这样一名核查工程师，他能够判断问题的性质和范围，同时能够提出相应的解决方案。请一名第三方核查工程师来做核查是一件很好的事情，因为他并没有直接参与所要进行核查的项目或设计，也就没有任何具有倾向性的偏见或想当然的行为。

核查这个步骤很简单。核查工程师应该审查设计要求和设计文档，并利用针对设计类型和复杂度而给出的特定核查清单，对照设计要求来验证最终设计。在版图规划阶段产生的核查清单就是所谓设计文档的典型例子。

记录所有出现的问题。在这里一个很好的想法就是，让电路设计工程师和负责将该设计集成到更高层设计的工程师参与到核查过程中。这样一来，若发现问题，我们就可以高效地实施提出的解决方案，并拥有所有相关信息。

一旦问题解决，标识出该问题的核查人员就应该取消该标识。

这个过程的最后一个步骤将生成一个经过参数提取的版图。

参数提取是指将设计传回给电路设计工程师的步骤。在这一步骤中，会将最终的适用于仿真的版本反标给电路设计工程，以做最终的后仿真。在后仿真之后，还可能会对设计进行一些调整。

参数提取也是根据版图自动生成一个适用于仿真的网表的过程，在这一网表中包含了与所有器件的连接关系、器件尺寸以及布线电容和电阻相一致的信息。

提取出来的网表是版图设计工程师与电路设计工程师之间的一种很好的交流工具。这一网表表明了版图设计已经完成，并正在等待最终的仿真结果

3.11.1 验证

对于最终文件的验证来说，需要考虑以下问题，

因为掩模厂商所要求的是GDSII文件，所以交付制造和进行最终检验的文件应是一致的。如果交付给掩模厂商的分层信息与设计中实际使用的分层信息不同，则最终的验证也应针对掩模所要求的GDSII文件进行。必须将数据库由在线使用和验证的设计分层转换为掩模厂商所要求的分层，在此基础上进行的验证才是有效的最终验证。但是，即便如此，还会有一些为了工艺需要而添加的结构无法通过DRC或者LVS验证。对于这些由结构引起的错误，绝对不能忽视，必须仔细检查，否则，就可能影响一些敏感的逻辑电路，并因此给我们带来一些问题。

使用由“冻结的”数据库生成的“冻结的”GDSII文件来进行最终的验证是非常重要的。这意味着没人能够改动原始联机数据，从而确保了数据的惟一性。

3.11.2 核查

在这一层次上，必须由富有经验的工程师来进行核查工作，而他至少应经历过一次或多次交付流片工作。作为一种基本的检查，版图设计工程师可以利用之前提供的核查清单，也可以增补任何与公司及工艺相关的问题。另一个重要的问题是核查那些不符合DRC命令文件的重新布局关键点（请参考核查清单）。

还有一项重要任务有时会被遗忘，那就是，如果DRC和LVS的命令文件中所验证的分层不是由版图设计生成的，而是在掩模制造之前由CAD软件生成的，那么就需要对设计进行重新核查。例如，将N型晶体管周围的N+层置于P+衬底上。通常情况下，这一层并不是由版图设计工程师绘制的，而是在掩模制造之前自动生成的。根据我们的经验，在掩模制造之前若没有对其进行检查，就很可能会带来一些问题。

3.11.3 交付步骤

所谓的交付步骤，就是指上述所有步骤，加上交付特定文件并发布数据的步骤。将每块芯片数据交付给掩模厂商时，需遵守公司的内部程序。例如，管理部门应和项目负责人以及电路、版图设计工程师一起，审查所有的核查报告和附加的公司标准交付程序，并对所要交付的磁带连同相关文档、数据一起签字后，再交付出去。

要养成一个良好的习惯，那就是留心对所有的宏单元、文档、验证结果、核查报告以及已交付数据芯片的相关命令文件进行归档，以便将来更易于发布。如果已交付的项目出现任何问题，也很容易查询到所有相关的记录和文档。

第4章　版图设计流程

在第1章中，我们给出了版图设计的定义，说明了版图设计在整个IC设计过程中所处的位置。在本章中，我们会为大家介绍“流程”的概念，“流程”这一术语将在全书中多次使用。

在这一章中，首先，我们将定义什么是流程，为什么会有许多不同的流程，所有这些流程是如何发展起来的，以及什么是当前最常用的流程类型。其次，我们将对当今最流行的设计流程进行实例研究。在本章的最后，我们将总结三个基本设计流程，以及满足这些流程需求的相应的工具类型。

4.1　什么是流程

流程的定义如下：

流程是一系列有效方法的集合，运用这些方法，可以实现和验证一个设计思想的有效描述，使最终的结果显示出预期功能的适当特性

所谓方法就是“如何”做某事。方法中不但定义了一个设计应该遵循的一系列步骤，也包含了需要知道的相关知识、职业技巧以及进行这些步骤时其他值得注意的地方。

所谓有效方法就是指该方法要适用于当前任务，这一点对于确保任何设计流程的有效性都是重要的。也就是说，我们所采用的流程必须符合所采用的设计方式的需要，否则设计必然会失败。

流程何时有效？

对于任一流程来说，最基本的要求是要能够“实现一个设计思想的有效描述”，这是一个需要理解的关键概念。因为仅仅将设计思想付诸笔端是不够的，它还应该是交流的工具。因此，描述设计思想时，所用数据必须是有效的，并且是设计思想的功能描述。电路图、方框图、网表以及版图制图的计算机数据库都是设计思想的有效描述，而这些有效描述正是所有IC设计的基础。

若连最基本的功能都无法实现，那根本就不可能真正拥有一个流程，更谈不上流程是否有效。但如果有两种流程都可以实现我们的设计思想，而我们又必须做出选择，那么该如何判断它们各自的有效性呢？

更加有效的流程应是这样的一种流程，通过这种流程，设计工程师能在最短的时间内，或者用最低的成本推出一款设计，而这款设计可以呈现出预期功能的适当特性。

这些特性包含了你在正式的设计规范中所预期的结果（见表4.1）。

总之，新一代工具供应商和新工具的分工更细、更专业，并设法逐一解决现有流程中尚未解决的一些问题，因此新的CAD工具种类迅速增加，但是毫无疑问，它们显然不能解决表4.1所列的所有问题，因此，版图设计对设计工程师的专业技能要求仍然很高。

表4.1　设计特性列表

行为	时序	功率	面积	接口
工艺	可维护性	可靠性	空隙率	流程兼容性
复杂性	抗扰性	成品率	电迁移	抗闩锁效应

"预期功能的适当特性"，其意思是，对于每个特性来说，我们的设计只需要"满足要求"，而不需要竭尽所能呈现出"最小或者最快"的特性，否则，过犹不及。比如说，如果我们只需要生产带后备箱的汽车，但却使用了生产赛车的设计流程，那么即使该流程可以使所有的性能指标都变得更好，也是一种无效的流程。

我们已经了解了何为适当性，那么，在执行流程的时候如何衡量设计的适当性呢？答案是，在执行流程的每一步中，都要不断地验证和衡量这些特性。

在竞争激烈的市场中，最终还是由诸如收入和市场份额等商业问题来决定产品的成败，这就是一个有效流程能够加大公司间差距的原因所在。有些公司会比其他公司更快地推出同类产品（因此可以抢在其他公司之前占领市场），有些公司会推出高性能的产品，还有一些公司会推出成本更低的产品（从而更加有效）。

总之，流程是关于应该如何实现设计的知识概括。针对特定类型的设计方式，设计流程定义了一系列的步骤和一整套的设计工具。成功企业能够使它们的设计流程的发展和优化与所交付产品的商业要求保持同步。本书将阐述在版图设计领域构成设计流程基础的基本原理。

过去，所有的设计都是采用全定制的方式来完成的，这意味着项目每一部分的设计、验证和版图设计都是用手工来完成的。但自从出现第一批计算机可以为IC设计提供合适平台和专用软件后，新产品的开发人员就着手修改他们的设计流程以适应于新的工具。过去，方法学和流程取决于设计类型（如全定制、ASIC、存储器），但现在，它们已经越来越多地成为了不同流程的一种混合和搭配。

下面我们将详细说明标准流程发展过程中最重要的变化，以及我们对流程不得不发展的原因的理解：

- **面市时间的压力：**一个完整的流程，即使只是一种全定制的流程，也比完全没有流程要有效得多。流程是一种可以促使设计更快、工具资源分配更合理的方法。重要的是，它可以缩短交付时间。比如说，设计一个10 000门的芯片，现在6个月内就可以完成，而以前则需要两年的时间。
- **芯片复杂度的增加：**即使不断地开发出更快的工具和计算机硬件，全定制的流程和工具也不能保证一个设计团队能够快速完成一个10万晶体管规模的设计。因此，芯片设计

工程师不但要研制新型芯片，而且要开发新的设计方法以解决设计复杂度的不断提高所带来的问题。

解决的方法是在更高的抽象层次上实现设计，并让设计工具可以处理更加抽象的模型，从而减少花在设计每个晶体管上的时间。在这一点上，基于硬件描述语言的设计方法学已经很成熟了，它使设计工程师能够在RTL级而不是在晶体管级设计逻辑电路。

- **设计服务和设计半导体公司数目的增加**：紧接而来的市场现实是，许多有经验的设计工程师决定离开他们服务的大型设计公司，转而经营自己的小公司。这些新兴公司通常会把注意力放在那些对于大公司而言太小或者不熟悉的市场上。

  这些新兴公司作为设计公司往往自己设计芯片，并按照现在称为是ASIC模式的一种商业模式，将其设计交由半导体制造公司制造生产。这些新兴公司需要并且会购买新型设计工具以及设计方法来支撑他们的业务。
- **芯片规模的增加和器件几何尺寸的缩小**：由于设计公司都在设法提高利润，并期望能在一块芯片上集成越来越多的晶体管，因此加工工艺也得到了不断的发展。在不到10年的时间内，晶体管栅的尺寸从5 μm减小到0.5 μm，而在随后的5年内，又从0.5 μm减小到0.18 μm。若器件中的晶体管使用这么小的尺寸，那么，一个微处理器芯片就能够集成10 000 000个晶体管，而DRAM芯片则可以集成256 000 000个晶体管。

  尺寸小的晶体管、电容和电阻意味着会产生许多模拟的和物理的效应，从而影响设计性能。因此，逻辑设计工程师现在不得不考虑这些以前只有模拟或射频（RF）/微波设计工程师才需要考虑的效应，强调信号的完整性以及版图参数提取工具的准确性是当今设计流程中所需重点考虑的问题。
- **片上系统设计**：巨大的芯片规模使设计工程师能够为了完成更有效的系统设计，而在一块芯片内集成许多不同的应用，这就是所谓的片上系统（SOC）的概念。除非有一个精于各种应用领域的庞大的设计团队，否则，在短时间内是无法用单一的设计工具或者平台设计出如此复杂的芯片的。

  买卖各种功能模块的新兴市场获得了发展，从而取代了由各个公司自己设计这些功能模块的状态，这些功能模块称为知识产权（IP）模块。买卖这类模块的商业手段促进了设计的可重用性，因此可以提高设计团队的设计效率。

  这样一来，一个设计团队虽需要开发整个芯片的技术规范，但只需设计那些属于他们专长领域中的模块。至于其他需要在不同层次上引入的模块，已经由其他设计团队或公司设计好并且进行过功能验证。

以上是反映设计流程中几个重大变革的例子，这些变革改变了人们设计VLSI芯片的方式。设计团队必须调整自己以适应市场和时间的压力、知识面的限制，以及不断改进的工艺能力；而新的设计流程和设计方法学正是不断成功的关键。

是什么决定了要使用的流程，或者说如何确定一个流程？答案是，所要设计的芯片类型和芯片上的不同模块决定了要使用的流程。对于同一芯片上的不同模块，使用的流程可能是不同的，对于SOC设计更是如此。微处理器需要一种能够及时完成设计的流程，而存储器则需要另外一种流程。通用产品是为公众所知的，因而我们将在后面几节分别讨论它们各自的流程。

4.2 微处理器设计流程

最著名的微处理器莫过于Intel公司的奔腾系列、Digital公司的Alpha系列以及工作站的RISC芯片。另外，还有其他小型微处理器，它们作为控制器件广泛应用于家用电器、商用机器、汽车、玩具等方面。

MPU或微处理器单元是一种极其复杂的集成电路，它能够接收编码指令，执行接收到的指令，以及发送要求执行的结果。

微处理器是最复杂的芯片，因此通常会明确地定义流程中的每一步，并在较高的抽象层次上对其进行检验。只有当这些方案经过证实后，才能进入标准的全定制设计流程。下面给出了一个被简化了的、针对微处理器的开发流程的实例（见图4.1）。

1. **芯片功能规范**：第一步是根据市场信息来确定芯片所要求的功能。基于芯片的功能模块和可能的应用，项目负责人会决定到底是采用已有的标准芯片架构，还是开发新的芯片架构。
   芯片必须符合总线接口、时序规范和制造封装的标准；因此，项目负责人必须要对这些要求的重要性和优先级进行排序。通常并不是每一个可能的要求都能得到满足，因为满足所有的要求需要花大量的时间。毕竟，如果芯片不能及时面市，即使功能完善，也不能给公司带来收益。
2. **芯片体系结构**：芯片体系结构实质上就是将芯片划分为功能模块、并将它们连接到一起的方式。在芯片体系结构的定义过程中，应选择和确定经过优化了的不同功能模块，以满足所要求的性能指标。
   就版图而言，体系结构代表着每个模块放置在何处，有多大，放置在何处才是合理的，它们之间的逻辑关系又是什么样的，在众多信号中哪些是重要的信号。在这个层次上，芯片看起来像是内部放置了许多空模块，并由pad框包围着。
3. **数字仿真**：数字或者功能仿真是作为验证打算设计的行为级模型的一部分来执行的。仿真验证芯片的体系结构是否可行，是否能执行期望的操作。
4. **电路图设计**：在这里，不同的功能模块被提炼成电路图，甚至详细到晶体管级。在这个层次上，设计工程师必须处理一些实际问题，比如说器件和连线的耗散热、功耗、电阻，以及电容。此外，还要验证设计的功能性和尺寸大小。

5. **版图设计**：当经过仿真的电路图交付给版图设计工程师来生成基于多边形的电路描述时，版图设计这个步骤就开始了。
6. **交付生产**：完成了整个芯片的版图设计，并将工艺要求和最终电路图进行了对比检查之后，就该是为制造厂商准备数据的时候了。

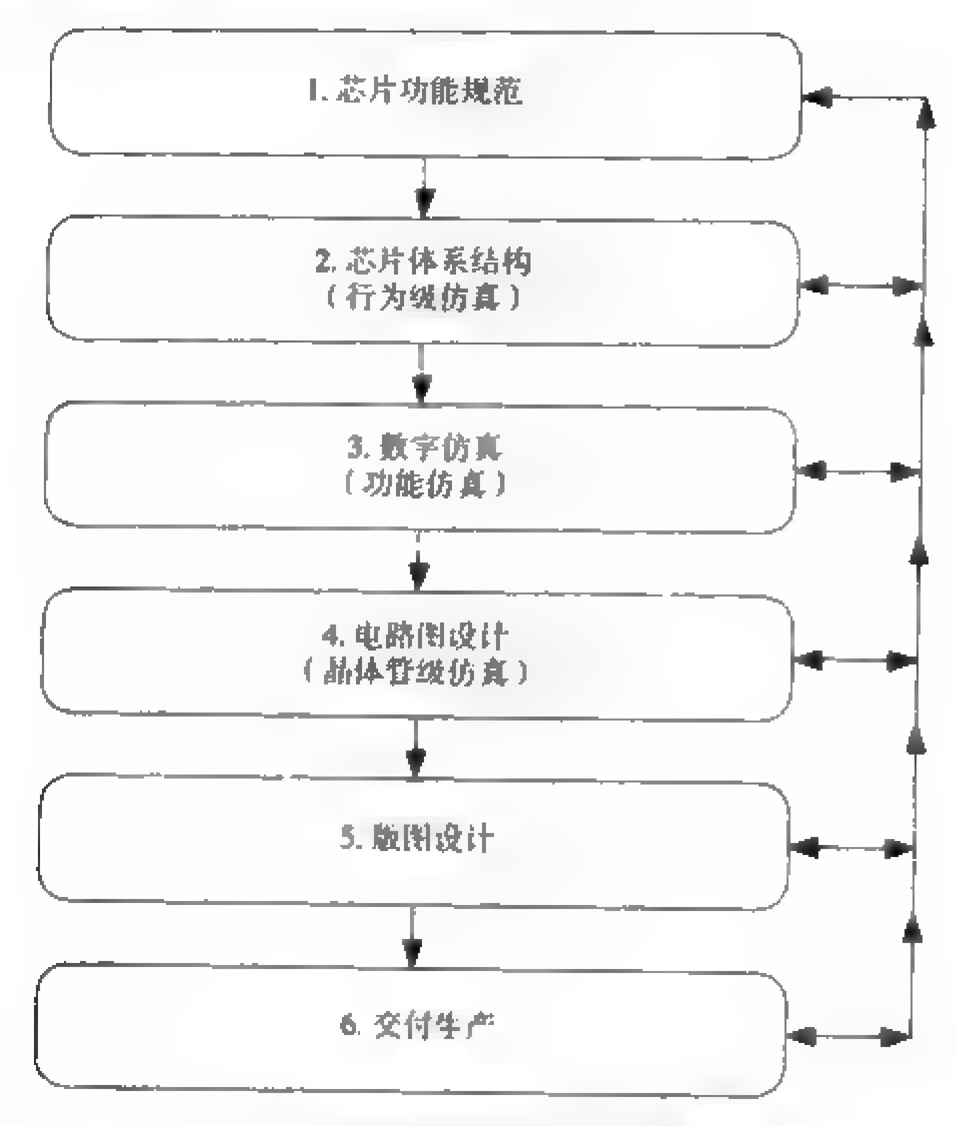


图4.1　针对微处理器的全定制流程

4.3　专用标准产品

不同于像DRAM这样的通用产品，专用标准产品（ASSP）是一种被设计成执行特定功能的标准产品。通常，大公司首先在VLSI设计领域取得主宰地位，之后，它们中的大多数开发出来的产品往往销量很大，但这些产品是标准产品，而不是专用标准产品。

而新兴公司却期望能在某一专业领域有所建树，或者立足于某一特殊应用领域，因此，通常会选择生产ASSP。如今芯片应用于生活的各个部分：电话应答机，汽车诊断电脑，全球定位系统（GPS），便携式电话，咖啡壶以及针对各种应用的电源。这其中的每一种应用都需要专用的芯片，新兴公司要在竞争激烈的环境中有利可图，就不得不开发新的设计方法，而这些方法又需要新的工具来支持。

不同ASSP的设计者会使用不同的设计流程，这是因为ASSP可以归于几种不同的类型，分别概述如下：

4.3.1 DSP

数字信号处理（或称为DSP），由数字电路来执行，这一数字电路用于处理信号接收和分析中的一类主要问题，而传统上这类问题是由模拟器件来解决的。在那些对超时稳定性有特殊要求以及对温度变化要求极高的场合，DSP正快速取代模拟信号处理功能。通过分析和传输波形（例如经由调制解调器通过电话线发送数据），DSP能用来放大、分析、过滤、调制或者处理标准真实世界中的信号，如图像、声音、雷达脉冲及其他信号。

依据设计的复杂性或者开发成本，我们可以按照ASIC的设计流程或者类似于微处理器的设计流程来实现DSP的设计，而后者主要是一个全定制的流程。

4.3.2 ASIC

与DRAM或微控制器等通用电路相反，专用集成电路（ASIC）是为了满足消费者特定需求而专门设计的半导体电路。

ASIC流程最大的挑战在于，设计是全新且专用的，没有任何经验可以用做建立体系结构的基础，因此，应着重考虑该设计体系结构的定义和验证。

ASIC流程的另一个特性是设计特别偏重于逻辑结构。对于超过5000个晶体管的ASIC电路而言，在全定制的环境下完成设计输入任务，实在过于繁重，通常不切实际。因此对于这类设计而言，基于硬件描述语言的设计方法是普遍使用的一种方法，这样设计工程师就可以完全不用考虑电路的晶体管级结构了。

就版图设计而言，情形是非常相似的。版图设计工程师不再需要关心晶体管级设计及结构，只需在模块级使用先进的布局布线工具就可以了。版图设计工程师甚至可能并不需要了解有关库的完整版图信息，因为在布局布线时，基于版图生成的屏蔽物就足以生成复杂的模块或芯片了。关于库的概念，我们将在下一章中进一步阐述。

现在，让我们来看一下ASIC流程是怎样工作的，以及要完成一个设计需经哪些步骤（见图4.2）。

1. **体系结构/行为级设计：**请见4.2节中芯片体系结构的定义。
2. **RTL级设计：**设计工程师开发并检验系统级和功能寄存器传输级的HDL代码，从而实现预期的功能。Verilog和VHDL是用于实现这种功能的标准语言。
3. **逻辑设计：**数字和功能仿真是作为验证预定设计的行为级模型的一部分来执行的。仿真验证芯片的体系结构是否可行，是否能执行期望的操作。

4 **逻辑/时序优化**：这一步骤是非常重要的一步，它使IC设计从基于电路图的全定制设计时代发生了彻底改变。如果前面步骤中完成的HDL代码不能综合，那么这些代码就是无用的。

在这一阶段，综合工具需要两个输入：用RTL代码描述的设计功能，以及用于综合的、带有时序信息的标准单元库。对于在HDL代码内编写的每一种功能，综合工具都会选择最合适的库单元或者库单元的组合来完成任务。综合的最终结果是一个包含标准库单元以及它们之间的连接关系的网表。

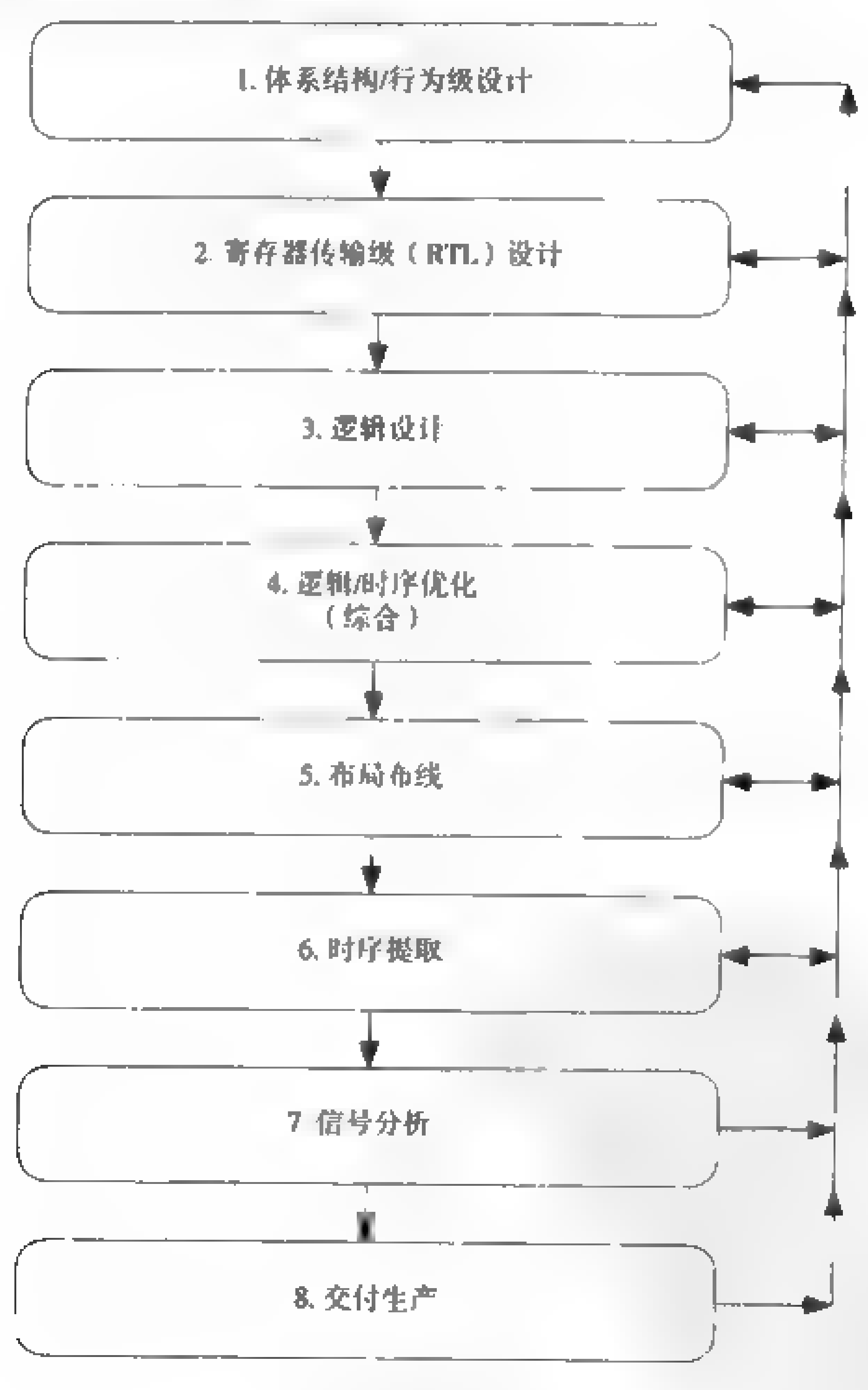


图4.2　详细的ASIC流程

5. **布局布线**：布局布线工具（P&R）是一种自动化工具，它需要以下的信息：
   - 标准单元库的物理信息，例如，单元尺寸、连接点、时序、布线屏蔽。
   - 综合过的网表，该网表详细列出了例化单元和连接关系，并在其中包含了设计中的约束和关键路径。
   - 针对连接层的所有工艺要求，包括布线层的设计规则、电阻和电容、功耗以及每一层的电迁移规则。

   利用这些信息，版图设计工具就能够自动生成版图，并依据最小面积和理想时序来优化版图。
6. **时序提取**：这一步是提取和计算在单元的布局布线完成后互连信号的时序。在版图实现后，所得到的延时信息会反馈给电路仿真工具来做设计的后仿真。提取的结果是一个标准延时格式（SDF）的文件。这个文件把从物理设计中提取的实际值反标到网表中去。
7. **信号分析**：现在，设计工程师可以使用这个新的网表对设计进行后仿真，从而检查它是否满足功能和时序的规范。一个设计通常要经过布局布线、时序提取、后仿真以及综合这几个步骤的多次循环，直到满足设计规范为止。
8. **交付生产**：一旦电路规范和版图设计规则都得到了满足，就可以将版图数据交付给制造厂商进行芯片制造了。

这就是在过去15年内得到了长足发展的基本的ASIC流程，由于设计复杂度的增长，设计工具和设计方法学也因此得到了改进。基本版图设计流程的改进包括以下几个方面：

- **不断提高的工具性能**：在布局布线工具市场中，所谓竞争力就是谁能在一次运行中对更多逻辑门进行布局布线。当芯片的规模增长到超过250 000个布局对象时，使用展平的方法效率就很低，因此布局布线工具市场转向层次化的设计。
- **将版图规划工具引入ASIC流程**：层次化的版图是使用版图规划工具来完成的，这些版图规划工具原先是全定制芯片设计者用来协调模块间的层次化需求的。

为了选择综合工具所用的单元，我们必须做出有关连线延时的许多假设。最初的方法是在布局布线之前使用互连线负载的统计模型。

当时序延时是100 ns时，由于布局布线工具的不同以及设计工程师经验的差异，有可能会产生正负10 ns的误差。当工艺达到0.25 μm，而关键信号的时序下降到10 ns以内时，就没有人可以承受10 ns的误差所带来的后果。因此，解决的办法就是尽早将一些物理信息带入ASIC流程中。

在芯片有实际的单元和互连数据之前，我们如何将版图数据引入到版图规划中？通过运行全局的布局算法和全局的布线程序，我们可以从顶层版图中提取出这些信号的布线负载信息。这些数据并不是百分之百的精确，但是它比先前所做的假设更接近于实际的情况。

若没有版图规划和全局布线，那么一个设计要经过5到10次迭代才能得到最终版图，若尽早利用版图规划和物理信息，那么每个设计的迭代过程能减少至2到3次。尽管它仍不是一个可以一次通过的流程，但是它能收敛得更快，从而使我们能更快地得到预期的结果。

工具之间会有许多联系，这一点将在第10章中详细讨论。就布局布线工具来说，建议使用联系紧密的版图规划工具、布局工具及布线工具，将这些工具紧密地集成在一起是十分必要的。

4.3.3 ASM

专用存储器（ASM）也是一种基于ASIC流程的芯片，但是它包含一个大容量存储器模块。如今几乎每一块芯片都包含存储器部分。它们可以是静态RAM、动态RAM、ROM等，但是通常其容量都不超过512 kb。而ASM则是一种要求有大量存储单元的芯片，容量可以从1 Mb一直到64 Mb，例如视频RAM（VRAM）、同步图像RAM（SGRAM）以及其他一些更复杂的设计。

在下一节中，我们将详细描述存储器模块的设计，它是采用全定制设计方式来设计的。如果存储器的设计是整个设计过程中的一部分，那么要实现芯片的设计，可以采用全定制设计流程，或者采用嵌入式存储器模块（IP）。大规模的ASM设计所采用的流程实际上属于4.5节中描述的片上系统设计流程。

4.4 存储器

现在市场上的存储器种类繁多。由于存储器用于通用消费领域的产量很高，因此它是功能模块中的重要商品之一。存储器从理论上说很简单：它的工作就是为了能够随时检索数据而存储数据。

存储器IC的实例有ROM、动态和静态的RAM、EPROM和EEPROM、SDRAM、FRAM和SLDRAM。若将这些实例都列出来的话，列表将非常长，这里只给出一个不同类型存储器的简短术语表。

- ROM（只读存储器）：存储器的内容是固定的，并且只能读出数据。
- SRAM（静态随机存取存储器）：只要芯片一上电，就能迅速运行并存储数据。
- DRAM（动态随机存取存储器）：与SRAM相似，只是密度更大且需要定期刷新，否则数据就会丢失。即存储的数据是动态的。
- PROM（可编程只读存储器）：通常只能编程一次，用做一次性写入、多次可读的器件。可以电编程和/或电擦除。

- HDRAM（高密度随机存取存储器）：一个用ASIC加工工艺实现的嵌入式存储器宏单元，它的优点在于不需要特定的制造工艺。MOSAID技术公司已经取得了多项可用于HDRAM的技术专利。

根据设计流程，存储器全是面积密集型的。存储器有效性的关键度量标准就是其密度和效率。

由于存储器的工艺通常要比其他的工艺领先一代，因此，它与工艺专家们紧密相关。由于存储器很大程度上依赖于工艺特性，因此在栅的最小加工尺寸方面，如0.25 μm、0.18 μm或0.11 μm，它必须领先。如今，如果我们能够生产出有一千万个晶体管的微处理器，那么所生产的DRAM存储器的容量就很容易超过256 Mb。

存储器的设计是采用全定制技术来实现的，这一事实表明了其设计将面积作为着重考虑的关键要素。在存储器设计中必须精确到每一微米！考虑到当前每块芯片中的DRAM存储器容量是256 Mb，存储单元面积的略微减小都能导致芯片面积大幅度的减小——存储单元减小的面积乘以系数256 000 000。

存储器的一个明显特征是许多单元具有重复的结构，这些结构在一个方向上是间距受限（pitch-limited）的。因此，根据设计流程，存储器的设计采用自底向上和自顶向下的设计方法，这点相当有趣。首先确定芯片和存储器的体系结构，然后从存储单元开始进行版图设计，逐步向外构建模块。芯片的尺寸主要取决于存储单元的大小以及它周围相关版图的大小，因此必须使用精细的全定制版图设计技术。

4.5 片上系统

当今最热门的设计流程之一是片上系统，或称为SOC的设计流程。如果需要设计一个规模庞大且复杂的芯片，而又没有足够的时间一切从零开始设计，使用SOC设计流程是最为合适的。

在这种类型的芯片设计中，关键词是**设计重用**和**知识产权**（IP）模块。谈到设计重用，应从一开始就带着重用的思想去设计每个模块，这种方法的缺点是：将每一个模块都设计成可重用的模块，它所花费的时间要比普通流程所花费的时间更长，其原因很明显：

- 规范必须涵盖通用问题，而不仅仅是当前项目中的某一特定问题。
- 必须完成更加通用的仿真。
- 若一个模块用特定工艺来实现，那么就不用为它定制设计结构。
- 要为工艺变化做额外的考虑。

然而，一旦我们可以在以后的设计中将模块作为一个核来重用，则就具有了许多显著的优点：

- 保存IP——设计IP的专家可能正忙于做下一个设计（或者已经离开了原来的公司）。
- 设计第二版或者第三版所需的时间减少了50%至70%，取决于所输入模块的种类（硬件、软件等）。
- 这些模块和方案已经过流片验证，使得下一个设计团队可以确信该模块是能够工作的。

就整个设计流程而言，SOC设计是一种混合流程，它依据每个特殊模块的类型将不同流程搭配在一起。在一个SOC设计中，可能用ASIC方式完成控制模块的设计，而用全定制技术完成模拟模块和存储器模块的设计。将所有这些模块类型集成在一起则正是SOC设计中的一个挑战。

现在让我们来看看如何确定一个能够满足SOC设计要求的流程。这个流程受许多因素制约，应在许多问题都弄明白了之后再确定，以下是需要考虑的情况：

- **工艺的选择**：将使用何种工艺？就嵌入式存储器来说，可能会使用混合的工艺。
- **装配芯片的板卡类型**：这时我们可能就要先确定好：封装的类型、安装尺寸（footprint）、芯片尺寸、价格、引脚位置和分配、功耗限制。
- **公司内部设计或者外部市场中由供应商销售的核/模块的适用性**：它们是软核还是硬核？
- **适用于选定工艺的库**：它们经过了流片验证吗？它们的特征功耗、特征速度是什么样的，是基于什么样的工具进行开发的？
- **设计中的可测性**：这是当今另一个热门的话题——可测性设计（DFT）。
- **制造工艺的限制**：时序范围、特殊层、光刻限制、封装限制等。
- **所选工艺的可靠性**：它是试验性的吗？是首次试验吗？是否经过几款能工作的芯片的验证？

就版图设计而言，所有工作中最重要的一项是，无论这些模块是内部设计的还是购买的，都应着手进行一个层次化的设计，其中明确地定义模块间的所有接口——即版图规划。但是，如果IP供应商没有考虑任何有效标准，如虚拟插座联盟（VSIA）所建立的一系列标准，或者提供IP的工艺不同，那么购买IP模块时就需非常谨慎。

模块可能是“软”核，这是因为它们并没有指定具体工艺，必须要用目标工艺小心地加以实现。如果是“硬”核模块，那么用新工艺时，必须在模块级将版图例化或者移入，提取版图参数并反标给仿真步骤来检查时序和功能

无论这里给出的流程多么新，到本书出版时，设计工程师、工艺专家、软件开发工程师还是会发现新的问题。必须记住的是，必须用创造性思维来解决日益更新的设计难题，而这正是趣味之所在。

4.6 CAD 工具作为流程的一部分

流程中的每一步通常都是通过一个特定的CAD工具来完成需要的操作的。在任一流程中，对工具的选择和掌握取决于许多因素，在本节中我们将讨论这些因素。

首先要理解的概念是，基本上所有的CAD工具都属于以下两类中的一类：

- **设计输入**：将设计思想实现成有效形式的一系列方法，该有效形式具有设计思想的所有期望特性。
- **设计验证**：分析并验证设计是否被正确地输入（例如，功能是否正确，能否按要求运行，是否是可制造的）的一系列方法。

不同的工具以不同的方式处理这两种要求，或者是解决不同的问题。正是这些不同的方法孕育了整个行业，在这个行业中，每个供应商都试图寻找出一种神奇的方案来帮助开发出最有效的设计流程。

随着时间的推移，可供选择和掌握的工具数目显著增加，这是因为在以下的各个方面都有了极大的发展：

1. **设计类型**：例如，微处理的设计流程与模拟元件的设计流程具有很大的差别，因此它们会使用一套不同的设计工具。在此例中，还需要将设计规模和复杂度的差异考虑在内，这使得两种设计需要不同的设计流程。
2. **实现方法**：为了使设计工程师能够更有效地实现他们的设计思想，一些设计工具可以在不同的抽象层次上运行。版图规划工具和多边形编辑工具之间的区别说明了这一点。它们都是版图输入工具，但却是在模块级（版图规划工具）或者晶体管级（多边形编辑工具）这两个不同层次上实现设计思想的。
   在电路输入领域中同样有许多不同的实现方法，电路图实现（schematic capture）是一种类型，HDL（高级描述语言，通常是VHDL或Verilog）则是另一种类型。电路图在晶体管级实现一个设计，而HDL则在RTL级（寄存器传输级）实现一个设计。
   这两类电路输入方式之间的区别会对版图设计产生巨大的影响，因为用这两种方式实现的设计在数据库格式、大小，以及最后得到的电路设计的复杂度等方面有着相当大的区别。基于RTL的设计无论在何处都可能导致多至数百万个例化单元需要进行版图设计，针对这种设计的流程和基于电路图设计的流程完全不同，后者将生成更小的设计。
3. **设计规模及复杂度**：有一类工具专门用来帮助版图设计工程师自动完成一些按理不可能由手工完成的工作。布局布线工具能够完成数百万个例化单元的布局布线，它就是此类自动化工具的示例。

4. **专业化程度：** 针对版图设计而言，每种工具仅集中于解决或者处理少量的必需特性。例如，在模块内部布信号线的步骤和在模块之间布信号线的步骤是分开的，并使用不同的工具，这是因为它们有着不同的要求和约束，用一种工具则无法满足所有条件。

   对于表4.1中所列的每一个特性可能都有专门的工具，因而产生了适用于那些需要处理所有问题的高级工程师的复杂流程。

5. **接口点：** 一个重要趋势是在设计流程中出现前馈环和反馈环。这就使得流程中的每一个步骤的接口点数量增多了。例如，版图规划工具给电路验证工具提供接口信息，同样也给其他的版图设计工具提供接口信息。而版图编辑工具也同样能够将它的结果反馈给版图布局工具。

   电路设计信息应该从电路设计工程师处一直传递到版图设计中去（作为约束或者目标），并且在每一步中，版图设计的结果应该反馈给电路设计工程师进行验证。如果这些信息的传输更频繁并趋于更高的精确度，那么整个设计过程将产生更好的设计结果。

6. **精确度要求：** 为了实现新的加工工艺，设计工具不得不进一步提高它们的设计能力和精确度。目前，参数提取工具已在精确度上得到了提高，例如3D视图解决了近体电容计算的技术难题。

7. **加速方法：** 通过使用以一种新方式来解决问题的新工具，常常可以加快流程中已有步骤的处理速度。该步骤所完成的功能并没有改变，而新工具中所使用的算法却加快了处理速度。一个典型的例子是，层次化版图验证方法要比先前的展平层次化方法快许多个数量级，并且仍能提供相同的功能。

8. **数据库格式：** 一些公司会为它们的工具开发独有的数据库格式。这些公司同样也会开发一些与现有的工具功能等同的工具，以便于在设计工程师使用独有数据格式时，提供给他们所需要的完备性能。另一种选择就是，这些公司可以开发转换工具，这种转换工具可以使其他数据格式转换为他们自行开发的数据格式。

   当要实现的电路来自于先前的设计或来自于第三方供应商时，同样也存在数据库的兼容性问题。在有些情况下，当通过重用或基于模块的设计利用他人的经验时，可能会影响到流程中工具的选择。通常会增加流程中的工具和/或步骤来验证和转换该电路，以便它与流程中当前使用的工具相兼容。

本章其余部分将通过特定的例子重点阐述这些发展对当前通用的、不同设计类型的流程的影响。设计类型通常决定所要使用的一系列工具，因为设计规模和复杂度决定了所要使用的工具，而这些工具会在合理的时间内完成设计。我们之所以要举例说明这些概念，是为了帮助读者理解这一系列适用于特定设计类型的工具。

4.6.1 模拟IC设计流程

当前绝大多数的进口IC芯片属于这类产品。模拟IC的功能范围从单个晶体管器件的简单功能，到以非常精确的操作特性为特征的复杂功能无所不有。运算放大器、转换器、锁相环只是这类器件的非常小的例子。

这些模拟器件被设计成能够展现精确的模拟特性，其输出电压电平和功耗必须小心地加以控制。由于芯片中的电路对晶体管特性的细微变化非常敏感，因此必须对其制造工艺进行精细的控制。因为这些器件会被大批量生产，所以比其他器件需要更彻底地检查其成品率和可靠性。表4.2概括了在模拟IC设计中最受关注的一些关键特性。

表4.2 模拟IC的关键特性

行为	时序	功率	面积	接口
工艺	可维护性	可靠性	空隙率	流程兼容性
复杂性	抗扰性	成品率	电迁移	抗闩锁效应

通常，熟悉和实现电路的基本行为功能相当简单。电路设计工程师只需花很少的时间来验证设计的行为功能是否被正确地实现，因为与其他类型的设计相比，模拟IC中的晶体管数目是非常小的。

那么，用何种流程来设计这种类型的IC呢？

答案是，这种流程可能是现存的、至今未变的最古老的设计流程之一。有一段时间，所有的IC设计都是用这种基本流程来完成的。

这种类型的流程通常称为“全定制”的设计流程（见图4.3），“全定制”这个术语起因于设计工程师对设计的所有方面有完全定制的自由。晶体管的尺寸和版图的实现需要仔细地加以设计，这样才能确保最终的设计能够完成预期的功能。

图4.3给出的流程简单明了。首先根据规范输入和验证电路；一旦完成，就进行版图设计、版图验证；最后是版图参数的提取以及后仿真，目的是确保设计的物理实现如设计的那样运行。

“电路输入”这一步又称为“电路图实现”（schematic capture），在此，输入晶体管和定制的逻辑门来作为设计的实现。工程师用手工来选择、放置以及连接元件。从流程的角度来看，电路图图形的美观固然重要，但是为了方便设计组开发、交流、调试设计的电路实现，一个简明易懂的电路图却更重要。

版图被绘制成一些多边形和线形。多边形编辑工具只能用来绘制单个的晶体管。在版图这个层次上，需要有非常详细的有关制造工艺和各层之间相互关系的知识。同时，全面理解不同的版图实现对电路性能的影响也是很重要的。

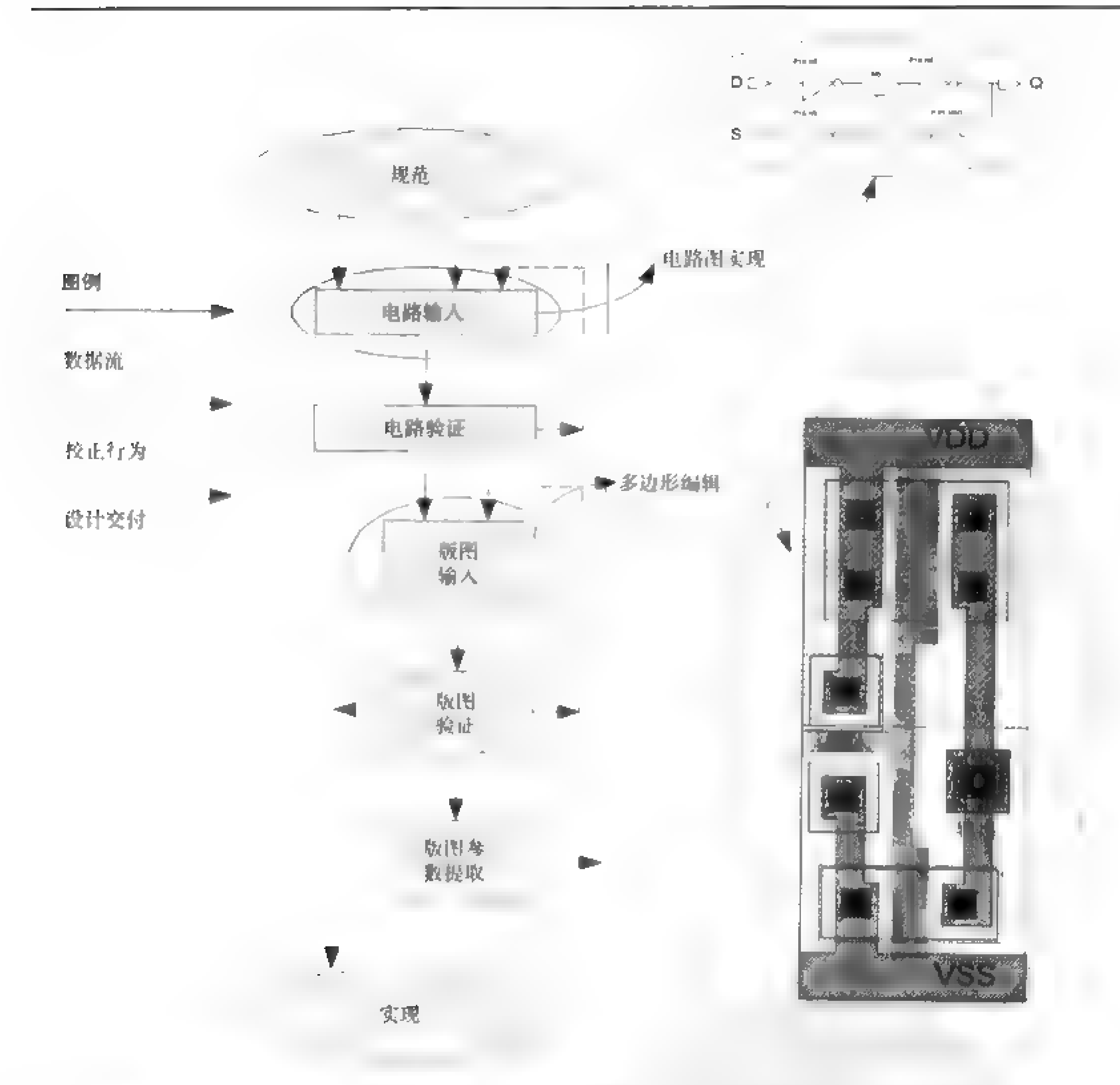


图4.3 全定制（或模拟）设计流程

电路图实现工具和多边形编辑工具构成了全定制设计流程的基础。这些工具就像我们在流程的定义中所说的那样，主要用于"实现一个设计思想的有效描述"。

要使这类流程自动化是相当困难的，因为设计组要考虑的主要问题，如版图完成之后其抗扰性、可靠性和成品率的评估，并不是利用现有工具就能够轻易解决的。电路验证过程通常还包括目测那些从复杂仿真模型中得到的大量的波形数据。

同样，用来确定设计是否"显示出预期功能的适当特性"而进行的大多数分析是一种劳动密集型的任务，需要有晶体管操作和制造工艺的物理特性等方面的专业知识。

总之，在全定制设计流程的实现过程中，模拟电路的设计既要求精确，又要求受到相应的控制，因为设计需要非常精准和严格的模拟性能特性。电路图实现工具和多边形编辑工具是该流程的基本工具。

4.6.2 ASIC设计流程

在IC设计产业，经常听到人们说“我们系统的核心是一个复杂的ASIC”或者“我们的设计是用ASIC方式来实现的”。设计公司招聘“具有ASIC设计能力”的人。这实际上指的是公司在寻找熟悉ASIC设计流程的人才。在这里，我们将阐述ASIC设计流程的工业定义。

从严格意义上来说，ASIC（专用集成电路）是一个通称，用来描述为某个特定应用而设计的器件，而不是作为多用途的器件。

例如，我们可以使用标准元件如计数器、逻辑门、触发器来设计一个无线通信系统。但是，如果使用ASIC元件，计数器和逻辑门会被集成到一个IC中，那么这个IC的功能只有在计划使用的相应配套系统中才有用。但在有些情况下，这些ASIC也可以作为多系统的标准产品出售。这种ASIC称为ASSP（专用标准产品）。

从理论上讲，任一ASIC都可以用前面提到的全定制设计流程（或者其他流程）来实现。然而在IC设计产业中，ASIC这个术语变得更像是某种设计流程而不是设计类型，尽管ASIC这个术语在两种情况下都可使用。

那么ASIC设计流程的含义是什么呢？

“综合”和“布局布线”。这两个IC设计产业中非常通用的术语准确地抓住了ASIC设计流程的本质。现在，它们运用的设计方法学已经彻底改变了IC设计的实现方式。

首先，让我们来理解该流程的基本原理，如图4.4所示。

下面是一些关键要点：

- **电路输入**：设计是用一种通常称为RTL（寄存器传输级）的软件语言实现的，但实际上这种软件语言是一种在RTL级编写的HDL（高级描述语言）。

  VHDL或者Verilog都是HDL语言的实例，它们可用来实现设计信息。这些语言支持许多不同的结构。用HDL描述一个设计思想的最通用的方式是，在称为RTL的抽象层次上编写代码。在这个抽象层次上，代码能够自动地转换成逻辑门和时序元件，例如触发器和锁存器。这个过程通常称为综合。

- **版图输入**：我们可以用布局布线工具对由综合工具生成的逻辑门和时序元件自动地进行布局和连接。

  应该注意的是，布局布线工具所生成的版图，可以“一次性构建即可保证正确无误”。因此在设计流程中，为后仿真而进行的版图参数提取步骤应在版图验证步骤之前完成。我们总是要求将版图验证作为流程的最后一步，一方面是为了确保版图数据库的完整性，另一方面是为了检查在对布局布线进行验证之后所做的任何增加或改变。

● **单元库**：ASIC流程的一个先决条件是单元"库"的存在。这个库包含综合和布局布线工具所要使用的逻辑门和时序元件。在完全遵循ASIC流程的任一设计流程中，整个设计的实现仅仅只要使用库单元就可以了。

ASIC设计流程可能是目前最流行的设计流程，绝大多数流程和工具的开发都以支持该方法为前提。

在一个技术水平发展迅猛、产品更新迅速的行业中，成功企业主要关注的是在最短的时间里推出最多的新产品。同时，尽可能缩短面市时间也是至关重要的。

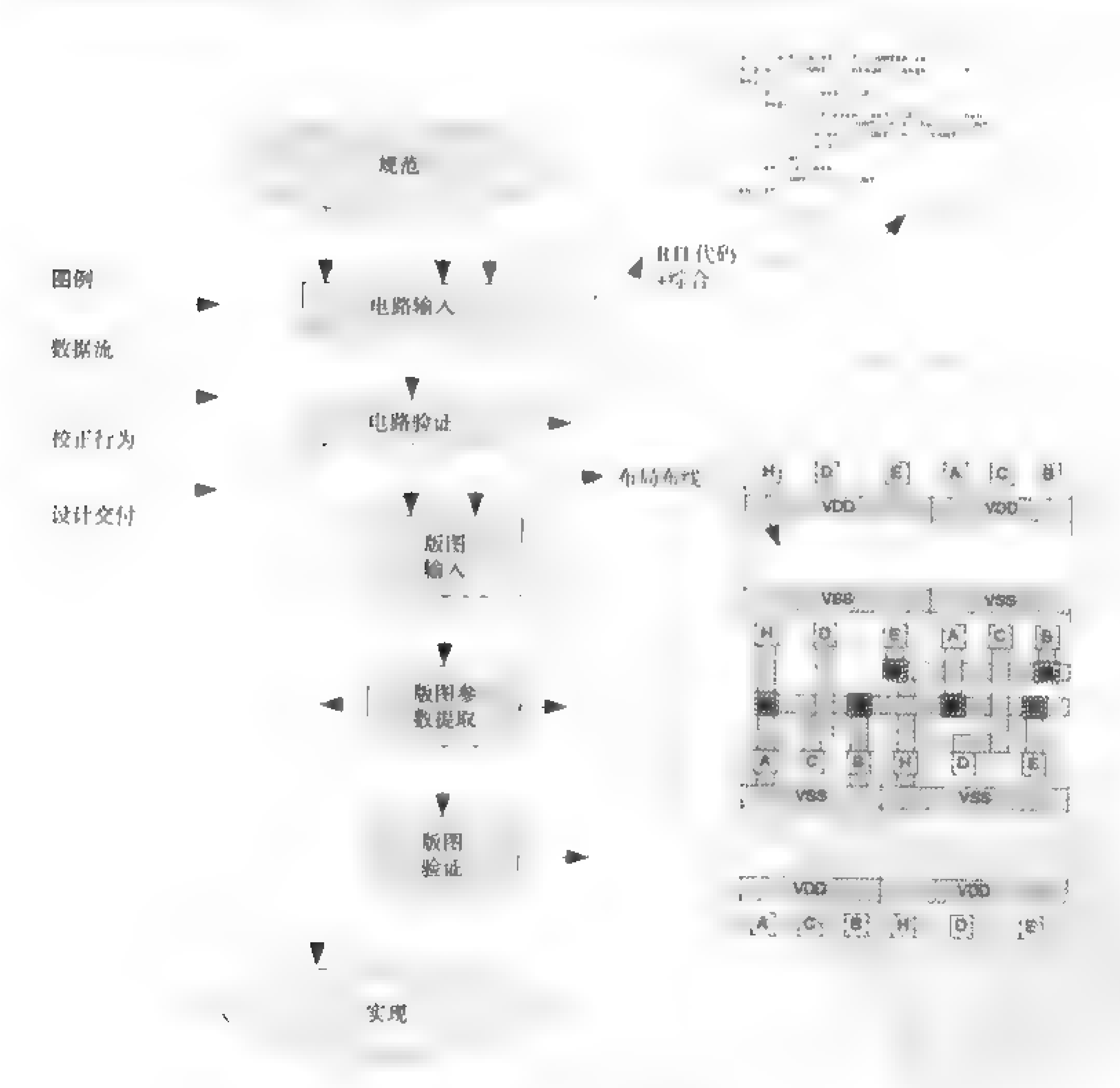


图4.4　ASIC设计流程

ASIC设计流程在许多方面都使IC设计的生产率发生了变革，尤其是和之前的全定制设计流程相比，更是如此。

在ASIC流程中，电路输入就是简单的"代码"，因此实现起来既快速又简便，在很短的时间内就可以实现庞大的功能。其最大的优点在于代码易于维护和重用，并且不依赖于某一特定的制造工艺。

图4.4中所示的代码样例是一个从0到9的计数器的VHDL代码（以每分钟输入25个单词的速度，2分钟就可以完成）。而绘制相应的电路图则会消耗很多时间。想像一下，在基于原理图的设计中（或在全定制的版图设计流程中），将计数器改成从100到0的递减计数器会花费多长的时间！

布局布线工具的发展使版图输入步骤最终得以与快速提高的电路设计效率相匹配。当电路设计是一个包含几万或十几万个例化单元的设计时，采用全定制的版图设计技术是不切实际的。

IC制造工艺的发展是构成AISC流程的所有自动化技术发展的主要驱动力之一。单个芯片中的有效晶体管数量随着时间的推移获得了显著增加。

由于设计过程已经提升到了逻辑门级，而在这一层次上，逻辑功能是关注的重点，因此，该流程中，即使器件数量不断增加，也都能得到充分的使用。在整个ASIC流程中，晶体管级和许多与工艺相关的因素都隐含在库和模型里。

表4.3概括了在一个ASIC流程中最受关注的一些关键特性。

表4.3 ASIC设计流程的关键特性

行为	时序	功率	面积	接口
工艺	可维护性	可靠性	空隙率	流程兼容性
复杂性	抗扰性	成品率	电迁移	抗闩锁效应

ASIC流程不仅提高了IC设计工程师的设计效率，而且吸引了很多人才进入了IC设计领域。电路和版图设计工程师之所以能从复杂的工艺设计规则中解脱出来，是因为这些规则隐含在了库里，电路设计工程师只需关心逻辑功能的实现、版图设计工程师可能只需要一些与布线层相关的工艺知识就足以完成版图设计了

换个角度看，ASIC设计流程就是使用HDL编码、综合，以及布局布线方法来实现设计的一种特定的设计流程。在这种情况下，用HDL作为电路输入是构成ASIC设计流程的基础。

我们可以推测出一个ASIC流程需要拥有的许多工具。HDL电路输入需要如下的工具和方法：

- 针对HDL代码的电路验证方法。
- 使用综合工具来实现HDL设计到逻辑门的转换

- 针对综合后得到的大型逻辑门设计的电路验证方法
- 用来将诸如分组约束和时序约束等电路约束前馈的工具。
- 用来将提取的版图数据转换成相应的电路验证数据的工具。

最终设计的规模和复杂度要求在版图设计中使用与全定制流程中不同的版图相关工具：

- 用于单元分组和单元导向布局的版图规划工具
- 以自动布局布线为形式的自动版图输入方法
- 针对大型设计的高性能版图验证和提取工具

此外，还有一些用于产生库的附加工具和方法：

- 用来给综合和仿真生成模型的特性提取工具
- 使用全定制流程实现版图单元、
- 或者，从现有库中移植单元
- 或者，用版图综合工具来生成库单元

总之，ASIC设计流程是一种最适合于实现复杂和大规模逻辑设计的流程 HDL电路输入、综合以及布局布线工具都是此类设计的理想选择

4.6.3　存储器IC设计流程

快速页面DRAM、高速缓存（cache）、EDO DRAM、SRAM和SDRAM这些术语对于那些刚买了个人电脑的人来说，应该不会陌生 任何个人电脑中的存储器的容量和类型肯定是商家宣传的重点，因为我们现在都知道电脑的存储器容量越大，性能就越好，速度就越快。

我们将在一个不同的小节单独论述存储器IC设计，因为这一设计类型的最好实现方式是使用“版图优先”的设计流程。就版图设计而言，它是少数几个在电路图设计之前就实现版图设计的设计流程之一！

为了说明这一点，我们将首先讨论一下存储器IC的结构。图4.5给出了一个关于DRAM存储器版图规划的例子。应注意的是，核存储单元及其辅助电路占用了芯片的大部分面积

需理解的一个重要概念是，一些相对来说是少量的叶单元差不多会被重复使用上百万次，因此，节省这些叶单元的面积对于节省整个芯片的面积是大有裨益的

这当中最重要的例子是存储单元本身。为了实现较小的尺寸，会对存储器单元进行一些特殊的设计，因此存储器制造过程是非常独特的 就存储单元设计而言，度量单位时常是纳米而不是微米。

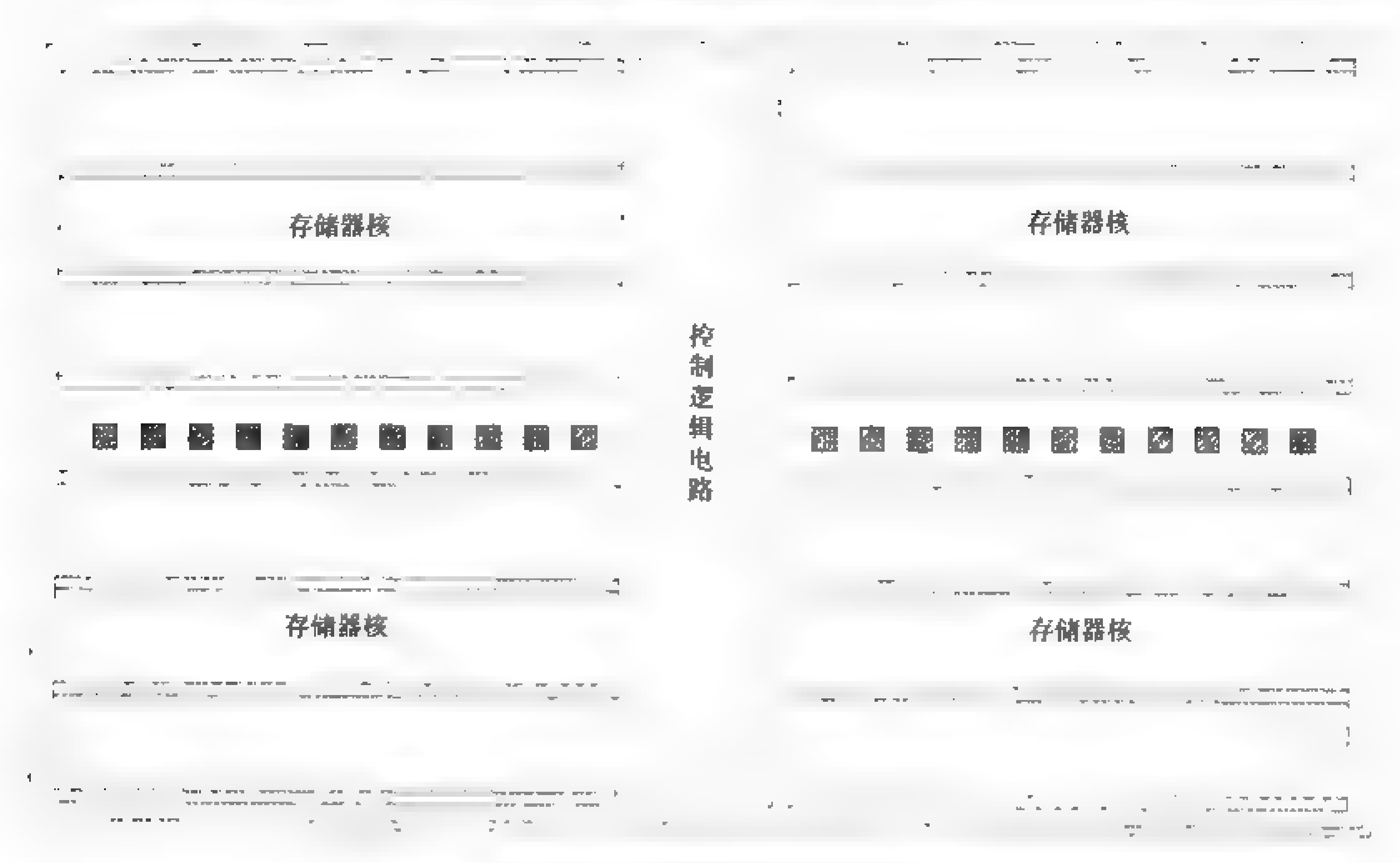


图4.5 DRAM版图规划示例

图4.6给出了在存储器设计中通常使用的设计流程。

以下是一些关键点：

- 对于存储器，关键的设计特性除了面积，还是面积。存储器IC是商品，额外的面积就等于额外的成本。
  存储器设计的目标就是在最小的芯片上装满一定数量的存储单元。因此，设计小组的目标是使外围和非存储单元电路所占用的面积最小化。存储单元的面积和整个芯片面积之比称为单元利用率。
- 如前所述，在存储器IC的设计中，首先完成的是版图设计。通常将存储单元和工艺一起开发，这样可以提供存储单元的版图。
- 这些存储单元构成了一个由内而外的存储器版图设计流程的基础。存储器核电路是从存储单元开始构建的。因为存储单元是间距受限的，所以解码器和灵敏放大器单元也是“间距受限”的。
- 存储器核的“间距受限”电路的设计流程是在版图和电路设计之间的一个非常紧凑的循环过程。设计工程师对许多电路设计进行折中，其目的是使最终设计能够高效地利用芯片面积。

这些设计折中也可以是结构上的。例如，可以精心地选择解码器和灵敏放大器的数目以确保得到一个合理的单元利用率。

- 一旦确定和验证了存储器的结构，通常在存储器编辑工具或版图平铺（layout-tiling）程序内绘制其结构以备后用。存储器结构非常规则，因此很容易被配置进入到一种自动化过程中。
- 我们仍然采用全定制的方式来设计存储器的控制逻辑，以使芯片的总体面积最小化。通常将电路图实现作为设计的实现方法。

总之，存储器IC设计流程中，面积是最优先考虑的因素，因此，在存储器芯片的开发过程中，版图特性是需要考虑的关键因素。正因为如此，存储器的设计是版图设计领域最具吸引力的挑战之一。

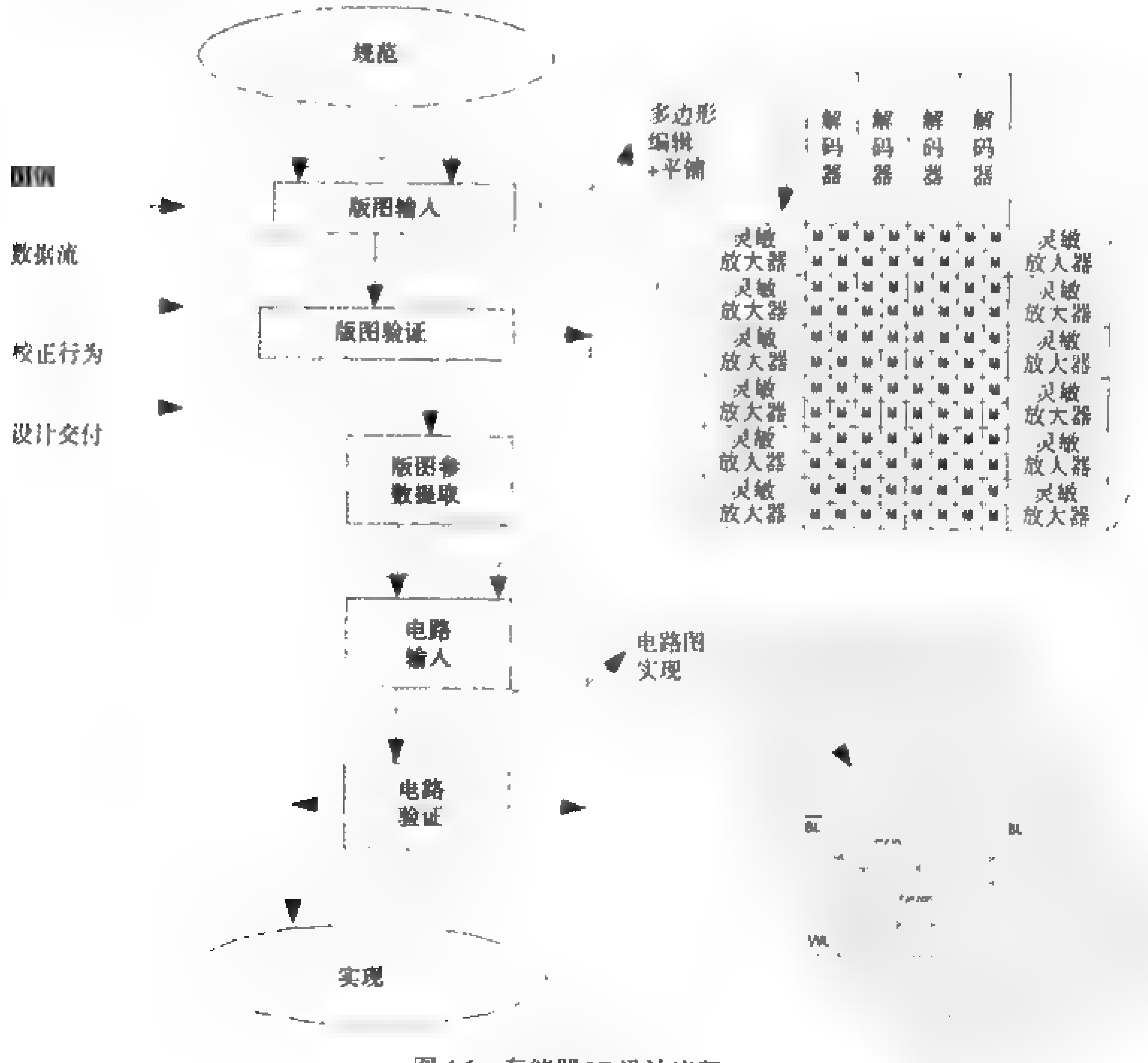


图4.6 存储器IC设计流程

4.6.4 微处理器和 SOC 设计流程

这类设计具有最为错综复杂的设计流程，它们代表 IC 设计技术发展的最高水平。由诸如 Intel 和 Motorola 等公司生产的微处理器的设计，结合了到目前为止我们讨论过的所有不同的设计流程。对于 IC 而言，SOC（片上系统）是统称，它是指那些将种类繁多的模块集成到一个芯片上的 IC 产品，这些模块的复杂程度不同，功能也不同。在这两种情况下，用单芯片的解决方案来实现所要求的功能无论对产品性能还是成本都是有益的，这就驱动了 SOC 的发展。

SOC 芯片的设计流程并不是单一的设计流程，而是全定制流程、ASIC 流程、存储器设计流程的混合流程。由于模块的不断发展，这三种流程同样也在变化。

要深入研究这些设计类型，不仅需要耗费惊人的工作量、拥有丰富的专业知识，更需要多年的积累。我们只需对这些 IC 中的某个芯片的版图布局进行简单的分析，就足以说明这些设计的多样性和复杂性了。每一个主要模块的布局和实现都是用最适合于它的流程来完成的。一旦每个模块都完成了，将所有模块组装在一起的单独流程也就完成了。

第5章　用于专用构建模块的版图设计高级技术

本章将介绍把版图设计知识应用到目前常用的不同类型版图子单元或“构建模块”的设计中。这些不同类型的叶单元，由于优化后可以作为基本的设计单元或库单元，在芯片的版图设计中重复使用，所以常采用全定制设计流程实现。

5.1　标准单元库

逻辑单元库是第4章中所讨论的ASIC设计流程中用到的构建模块的集合。由于这种库具有通用接口实现和规则结构，所以通常称为“标准单元”库。

单元库提供了用于综合的功能构建模块，同时为布局布线提供这些单元的版图描述。值得注意的是，硬件描述语言（HDL）综合的过程将所选择的逻辑单元限制在了那些库中所提供的单元中。这也确保了在使用布局布线工具进行设计时单元的物理或版图描述一定存在。

5.1.1　标准单元简史

掌握标准单元所要求的版图特性的一种方式，是去了解它们的发展史以及隐含于发展中的深层次原因。一旦我们掌握了设计过程中隐含的概念和方法，完全领会标准单元本身的版图设计要求也就会变得更加容易。

开发出标准单元库的原因是：

- 对于全定制设计来说，独立模块的规模变得过于庞大和复杂，因此就存在着加快电路和版图设计过程的需要。
- 缺乏具有手工实现复杂全定制模块设计能力的专业人员，而自动化工具缓解了这个问题。
- 典型加工工艺的进步，包括布线金属层从1层金属增加到2层金属或3层金属。对最佳结果的实现，这进一步增加了全定制版图设计过程的复杂性。
- 甚至在全定制设计流程中，当构建模块以预定义的标准来实现时，规模在20个单元以上的布局布线会更加容易。而单元接口的标准化在库中就可以实现了。

这一问题的解决方法是通过使用预先定义好的、特征化的“构建模块”（单元）来简化大规模数字集成电路的电路和版图设计。

在电路综合工具应用之前，一开始的想法只是设计一些预先定义的简单逻辑电路，比如反相器、与非门、或非门以及触发器等。这些简单的逻辑电路由一位专家来设计并进行分析，然后在项目中使用。每个人都可以采用这些逻辑电路来作为他们的电路中的构建模块。

为了使一种特别的逻辑单元可以应用于不同的场合，将库进行了扩展，使其包括了每种逻辑单元在各种尺寸下的实现。起初，不同的设计尺寸是由个别设计工程师随意决定的。

进一步的改进是在库中为每种逻辑单元定义了不同尺寸的实现，以便于使设计更易于"一次性构建即可保证正确无误"。为了驱动越来越大的负载，对信号进行放大应遵守扇出原则。例如，如果最小尺寸的反相器定义为P2.5/N1.25，那么单元库中不同反相器的尺寸都应是这个尺寸的倍数。2 × 反相器或INV × 2的尺寸应为P5/N2.5，INV × 4的尺寸为P10/N5，依次类推

尺寸的标准化确保在电路设计阶段不会使用其他非常规的尺寸，比如P8/N4或P6/N3等，并且可以最大限度地利用已实现的组元。

当综合流程发展之后，电路设计工程师实际上不会接触真实的版图单元，对标准化的需求也因此愈加强烈了。正如前面所提到的，综合工具在综合的过程中自动选择最合适的单元；因此，如果可选择的单元更多，综合工具就更可能对电路进行优化。

影响库单元发展的另一个因素是第一代自动布局布线工具所带来的影响。第一代自动布线工具的出现开始改变设计工程师实现全定制连接的方式，因为当布线工具利用以某种方式构建的单元来进行布线时才可以达到最佳效果。而标准单元的设计一直受到自动工具限制的影响。

如今，标准单元已经成为ASIC设计的基础。而一些设计公司的专营业务就是设计工艺库，以及把工艺库移植到不同的制造工艺中 许多EDA厂商也专门为库单元提供电路和物理设计工具。

虽然标准单元主要用于专用集成电路（ASIC）设计，然而这一设计方法同样也广泛应用于实现全定制设计中的"不规整逻辑"。最初，一个电路被分成了若干个小模块，每一个小模块都等效于一些预先定义的功能模块。在每个逻辑模块内，单元的实现来自于一个库单元集合。一般说来，这种库要比商用的专用集成电路（ASIC）库小得多，但是使用方法是相同的。

5.1.2 标准单元特性

图5.1给出的是一种简单的与非门（NAND）单元。我们可以看到一种逻辑门的标准单元描述。前面已经讨论过，ASIC版图设计是在单元级或门级实现的，而门级单元的内部详细信息是不要求掌握的。

在这一节中，我们将讨论单元本身的设计，其目的就是理解标准单元的设计以使其与ASIC设计流程相兼容。这里有许多问题需要考虑。然而，首要的一点就是单元库应该与所用的制造工艺的特殊要求和特征兼容。

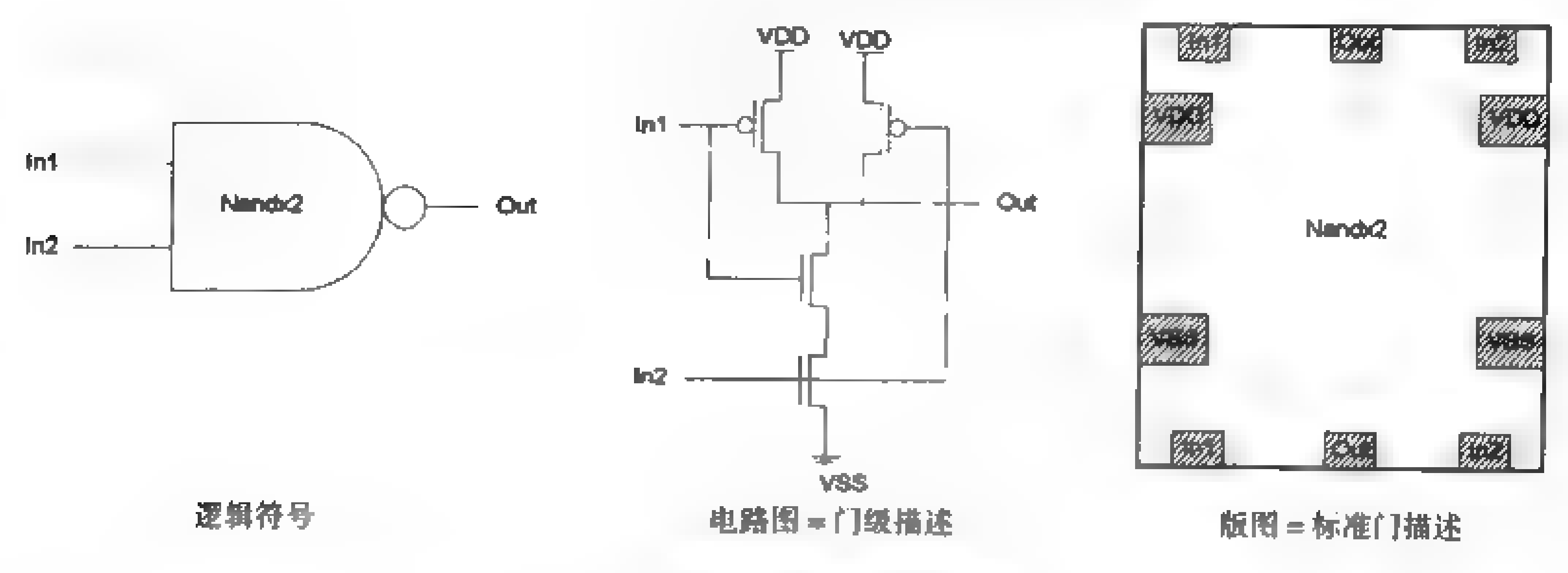


图 5.1 标准单元与非门实例

通常，必须依据目标制造工艺库中可用的布线层数来选择标准单元的设计或结构。在一些特定情况下，单元的设计还取决于所提供的金属层的特性。

下面列出了所有标准单元库共有的特性：

电路设计相关特征：

- 每个单元的功能、电学特性都要经过测试、分析和说明。通常会先生产一块测试芯片，然后通过实际的硅芯片对每个单元的性能进行分析。有时仅仅完成一个工艺特征步骤来生成晶体管特性的仿真模型，而库特性分析工具就是使用这些模型来建立每个单元的仿真模型。
- 为每种单元类型设计多种驱动强度的实现。而且，不同的驱动强度都是基本尺寸或最小尺寸的倍数。

标准单元基本形状相关特征：

- 在标准单元版图设计期间，用预先定义的模板建立单元，以保证满足所有的要求。模板应该包括单元的高度、阱的布局、N 型晶体管、P 型晶体管和一些要遵守的准则，以此来确保单元能垂直或水平地翻转，而且当其被放在其他单元旁边时不致引起 DRC 等规则错误。
- 所有单元都是矩形的。
- 对于特定的行或芯片区域，所有单元都是等高的。一个库可能包含很多种标准单元的集合。例如，不同单元可用于逻辑、数据通道和 I/O 接口等。
- 每个单元长度是由粗栅格的倍数构成的，这些栅格由下列因素决定：
  特殊的设计规则（例如最小的阱宽度）。
  满足更为简单便捷的布局的需要（使用栅格减少了可能的布局坐标的数目，从而使布局过程的速度加快）。

- 对整个库来说，电源线要有预先定义的宽度和位置——在整个单元长度范围内，电源线的宽度总是一致的。

单元接口相关特性：

- 所有输入、输出端口都拥有预先定义的类型、层、位置、尺寸和接口点。这些特性由用于实现设计的布局工具和/或布线工具来决定。端口是为布线工具布线准备的，应根据布线工具对其进行优化，以获得最好的结果。
  例如，通过使用定义在栅格上的信号间距可以使布线变得更加简便快捷。栅格技术的运用简化了布线工具的算法，所以，布线工具将占用较少的计算资源。
- 单元接口设计可以共享一些连接。例如，连接到电源的晶体管的源端连接可以共用。在一定的条件下，单元间可以共用衬底和阱接触孔。
- 矩形形状和对每个布线层的显示屏蔽也是每个单元的特点。针对每个布线层可以分别定义显示屏蔽，也可以把整个单元都做显示屏蔽。显示屏蔽可以是任何形状，它们不局限于矩形，但必须能被布线工具识别出来。
- 所有的不能共用的多边形和单元边界之间的距离必须等于设计规则中层间距的一半，以确保相邻单元连接时结构的正确。

单元库方面需要注意的是：

- 没有包含晶体管的单元，称为填充单元（feed-through或filler cell）。当单元上没有更多的布线资源时，可以将填充单元添加到单元间以允许垂直连接。
- 对于I/O单元，芯片的拐角，即单元两行交接处，有一些特殊形状的单元。

一个典型的标准单元库有几百个单元，而高级的库则有1000个以上的单元。有些单元库会针对低功耗、高速和高空隙率而进行专门设计、开发。

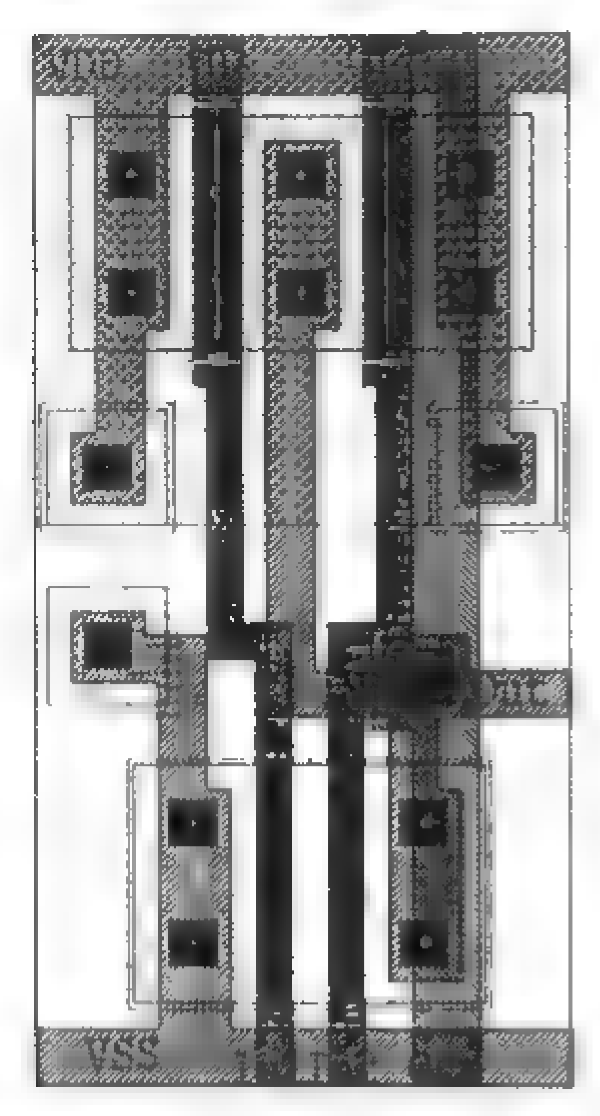
5.1.3 标准单元结构

正如上一节所提到的，标准单元的设计和结构依赖于加工工艺中可用的布线层数。让我们来看看这是为什么。图5.2给出了一个标准单元的例子。

注意，在单元内部仅有一个分层用于形成大多数的晶体管内部连接。这个标准单元的版本与早期的通道布线工具是兼容的。这些布线工具仅能连接放置在单元边界上的引脚。

从图5.2可以看出，与单元端口连接的第二层金属只是连接单元的顶部或底部。

图5.3所示的设计由两行和两行间的布线通道构成。值得注意的是设计中的填充单元。这些单元是空的，只提供垂直方向的通道，用以连接不同的布线通道上的信号。



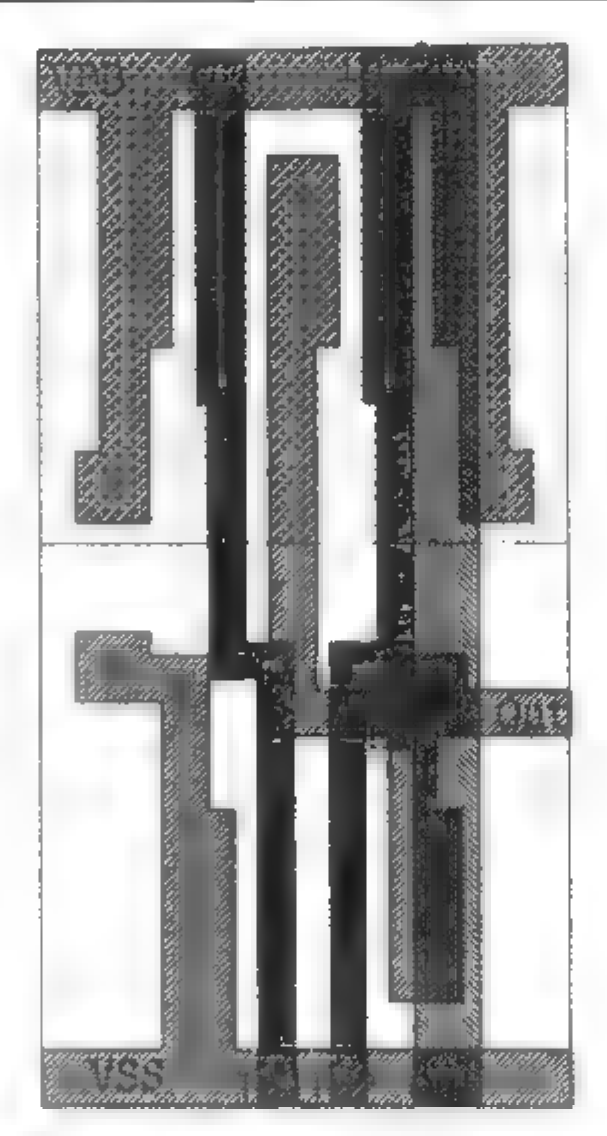


图 5.2　显示所有层的标准单元实例

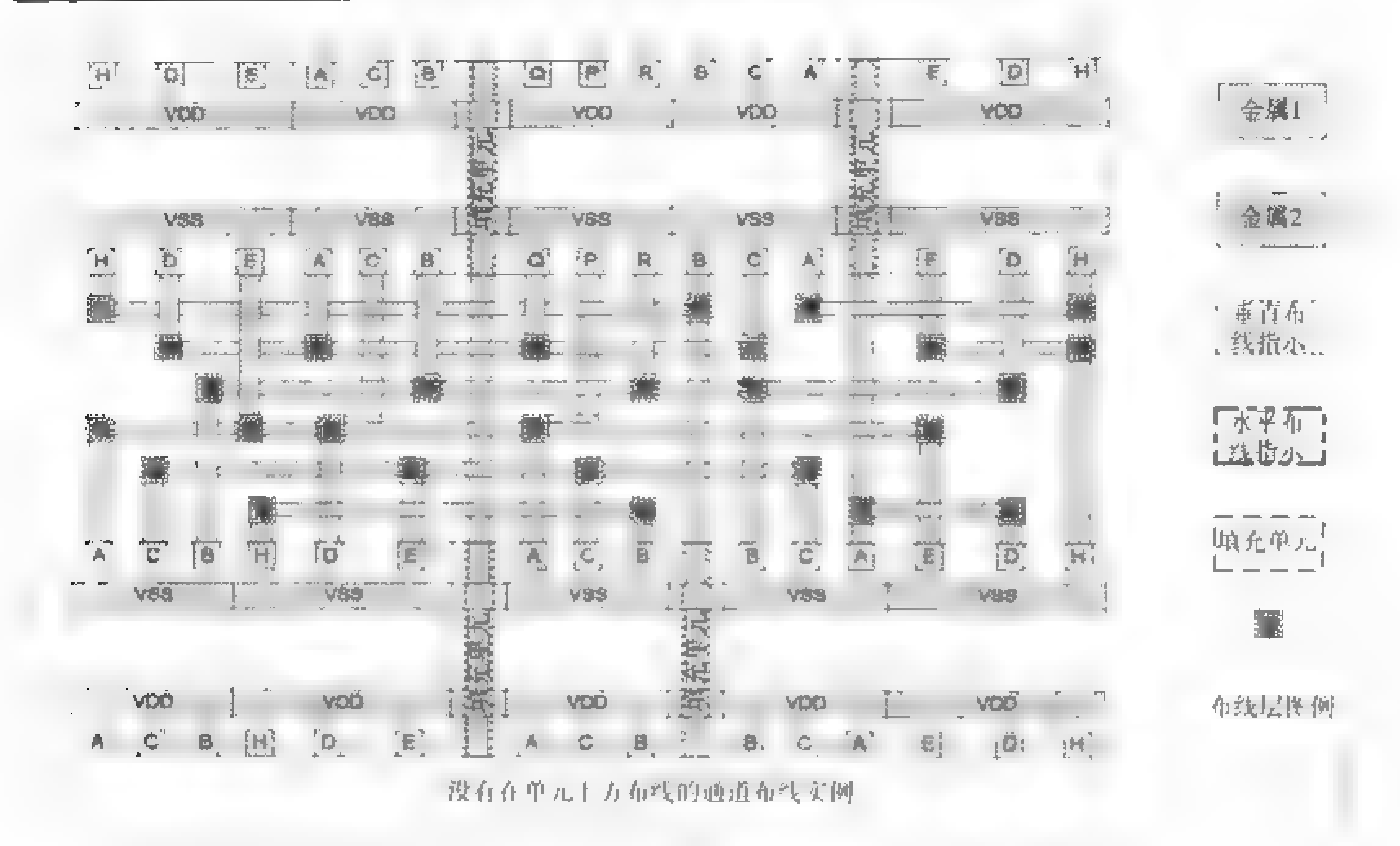
对于仅提供两层布线的工艺而言，填充单元是添加垂直布线通道的惟一方法。在这种情况下，布线是分两步完成的。首先，使用足够的填充单元来完成一定数量垂直方向的信号的通过，接着完成其他的布线。

当单元中间含有通孔的设计完成后，可以把端口放在中间，在单元上方进行布线，请参考图 5.3 所示的例子。这时，垂直布线通道可能位于单元上方，并且在最终设计中减少了填充单元的使用。

图 5.4 比较了两种单元结构以及设计中端口和接口的不同。

注意在例 A 中，设计的端口靠近单元的顶部和底部，以确保有效地使用布线通道，并减少分离的填充单元的使用。通过单元顶部和底部两边的端口，在单元上面走线有着重要的意义，因为它增加了信号的寄生负载并减少了最终设计的整体空隙率。而例 B 和例 C 就不会受到在单元上方走线的影响，并且与例 A 相同的单元的整体面积可能会更小。

图 5.4 的例 C 展示了在单元上方布线的优势和巧妙的接口设计。在这种情况下，如果与其连接的单元和它是相邻的，则端口 IN1 和 IN2 的连接会很短，而且不会浪费任何布线通道的间距。



没有在单元上方布线的通道布线实例

图 5.3 通道布线

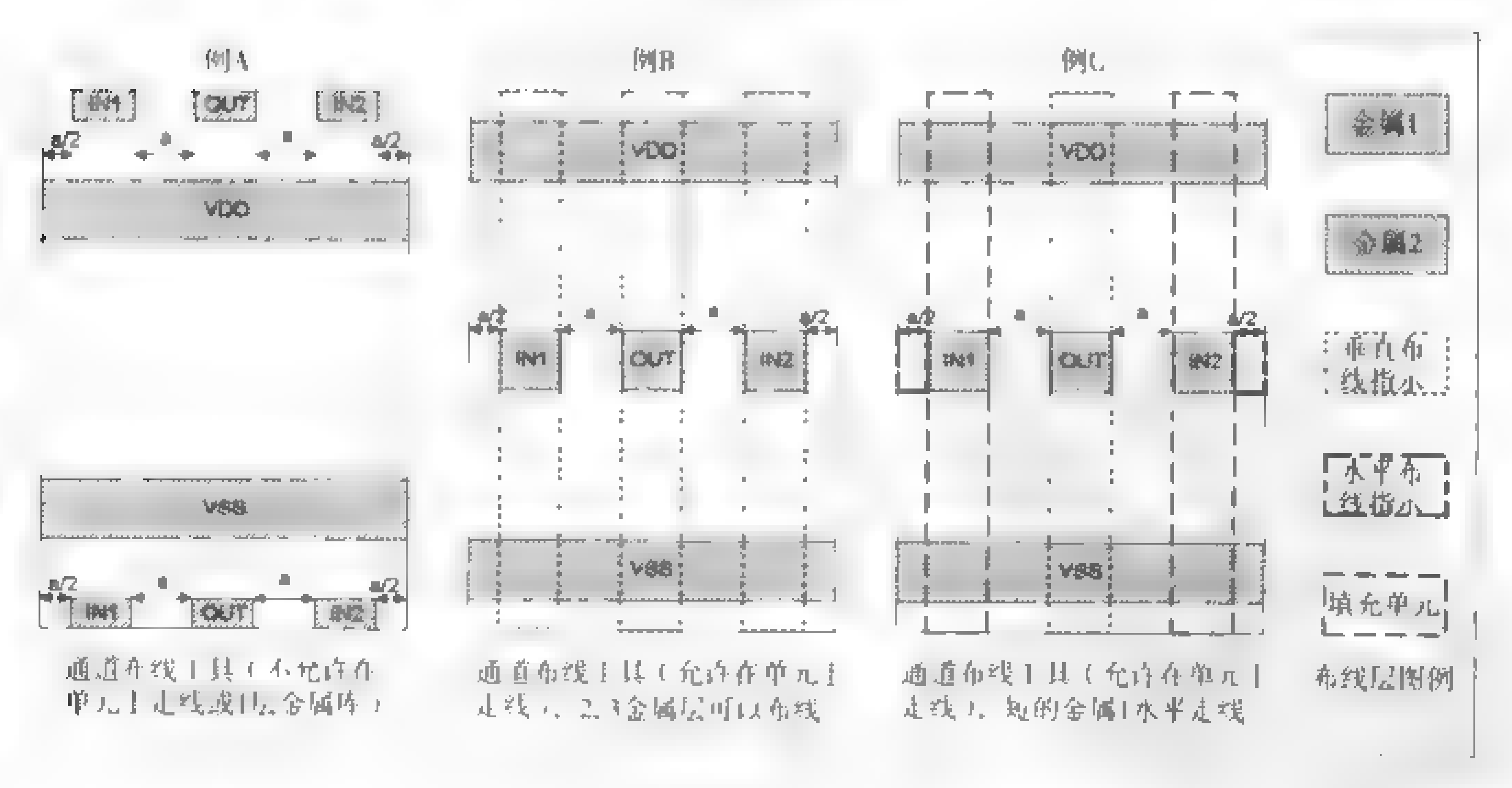


图 5.4 单元接口的比较

图5.5比较了两种布线的设计，一种使用了在单元上方布线的方法，一种没有使用在单元上方布线的方法。两种设计都只支持两层金属布线，而区别在于是否可以在单元上方布线。

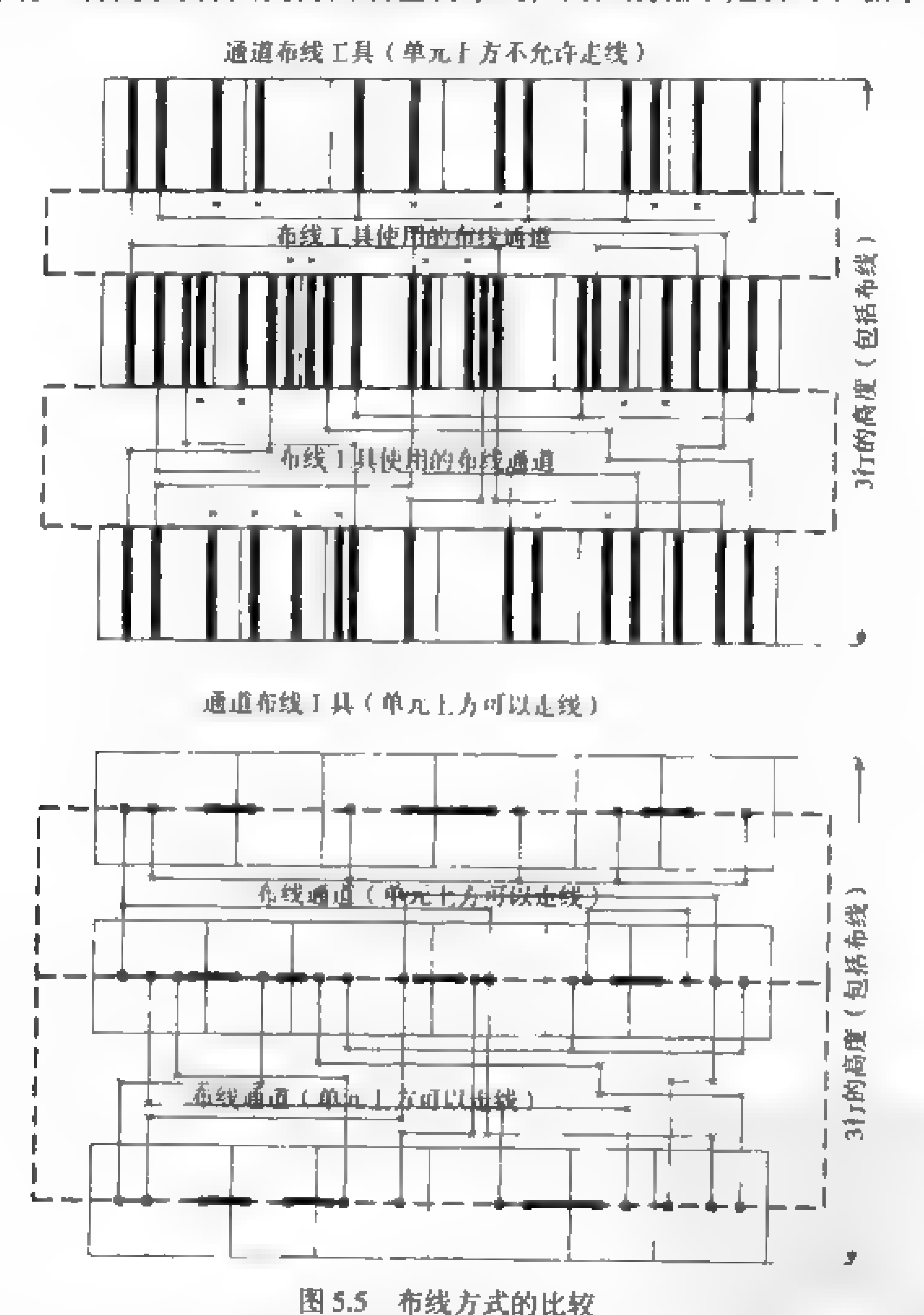


图5.5　布线方式的比较

如图5.5所示，在单元边界之间用做布线的通道只能是零间距。第二个例子中的限制就不那么严格了。第二个设计的面积明显小多了，而且对于规模较大的设计区别会更大。采用在单元上方布线面积较小的原因如下：

- 布线工具可以在相邻的输入-输出端口连接之间进行水平布线。这是节省面积的主要方法，因为单元间的布线通道由于这一影响而减少了。

- 可以使用的通道尺寸包括了单元区域，因此利用率更高。
- 由于相邻单元间金属1的水平连接，也由于省去了在单元顶部和底部内侧之间端口信号的布线，而导致单元尺寸减小，从而使空隙率变得更大。

在3层金属的布线工艺中，几乎所有的通道都能除去，所有的布线都可以在单元上方完成。

5.1.4 标准单元的其他相关概念

前面已经提到过的术语：设计的“空隙率”（porosity）。设计的“空隙率”可以作为衡量设计质量和/或效率的标准使用。在版图设计中，“空隙率”的定义为：可全部用于布线的面积之于整个单元面积的比率，其中布线面积不包括单元的引脚部分。

例如，当使用很多布线通道的时候，“空隙率”很小。理想情况下，除去布线通道或仅用实现设计所需的逻辑单元就能增加空隙率。

为了保证在合理的时间内完成设计，充分利用布局布线工具的优势，我们需要在库中对空隙率做一个折中。同时也需要避免这些布局布线工具可能拥有的限制。在设计单元库之前，熟悉布局布线工具的手册是非常必要的。

另外，前面提到过，布线和单元尺寸格点可以减少布局布线工具所占用的计算资源。在设计标准单元时，值得注意的问题如下：

- 选择布线格点很重要。目前使用的格点有三种：行到行、行到通孔、通孔到通孔。图5.6说明了这三种格点以及它们对标准单元设计的影响。

  如果采用行到行的间距，那么在使用通孔时很多金属走线需要拐角。这时，单元的尺寸是最小的，但是需要牺牲可布线性，并且布线工具完成布线所需的时间也更长。

  同样，如果采用行到通孔的间距，当两个通孔放置在相邻的线上时，金属走线也需要拐角。而尝试避免这些拐角的出现则会引起版图资源的消耗，可能会抵消掉信号间距缩小所带来的面积节省。

  在采用通孔到通孔间距的情况下，设计的空隙率最大，且布线更简单。
- 对于库而言，标准单元高度和最大单元长度将由使用的工具、库的用途、空隙率和布线层的数目决定。

  例如，当使用金属1作为单元内部连线的层时，就有这样一条设计规则，“单元的最大长度不能超过宽度的6倍”。当单元长度变长时，宽度也要增加（可能是两倍）。这是因为合理的宽长比可以使布局工具的工作更容易。长的单元更有可能使用金属2作为内部跨接线，从而使得布线空隙率降低。
- 在布局布线前，需要注意整个模块的电源规划问题，进行标准单元设计时，总是先处理电源的绑定和连接。需要考虑以下问题：

标准单元间的互连
标准单元内部布线通道（track）
与通道宽度和通孔数目相关的电迁移
电阻

- 因为布局布线工具仅仅与标准单元的布局和连接有关，所以考虑标准单元的提取很有价值。标准单元提取的不是设计的物理部分，而是布局布线工具用到的部分，它由下列几个部分组成：
  单元边界
  目标引脚的位置和形状
  布线屏蔽（obstructions）

布线屏蔽是在单元指定层内不允许布线的区域。例如，大多数标准单元里只有很小的区域允许金属 1 布线，因为金属 1 多数用于单元内部连接。

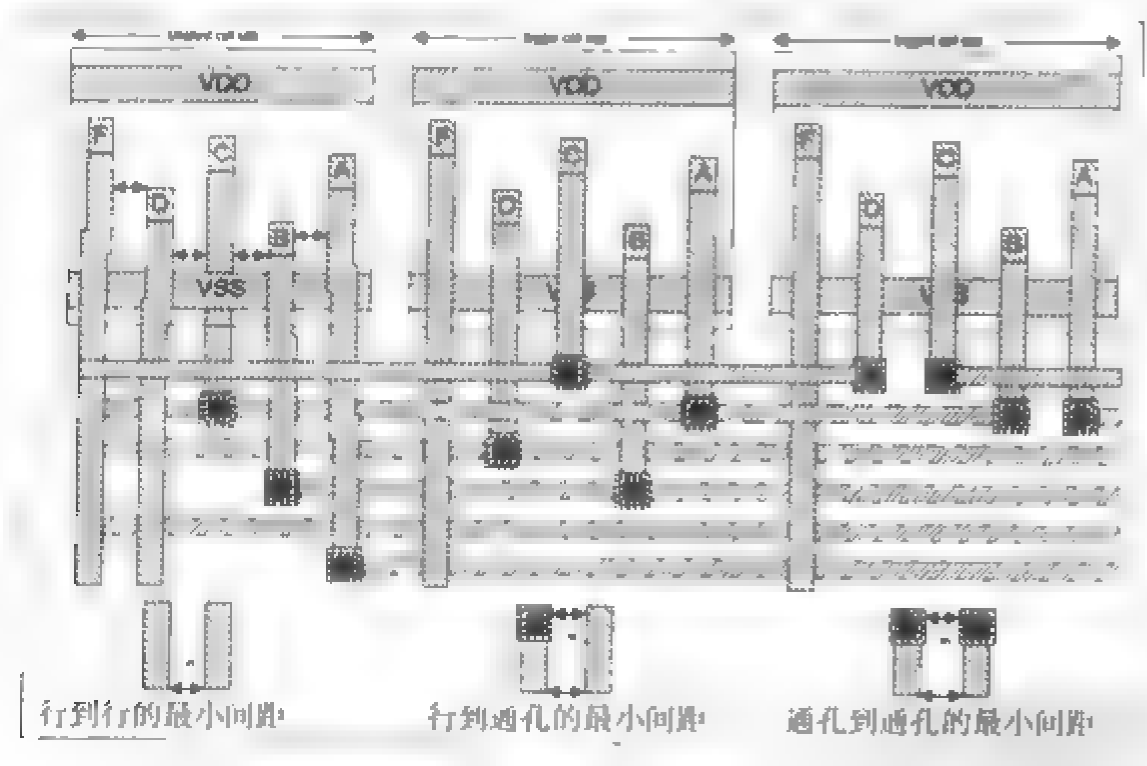
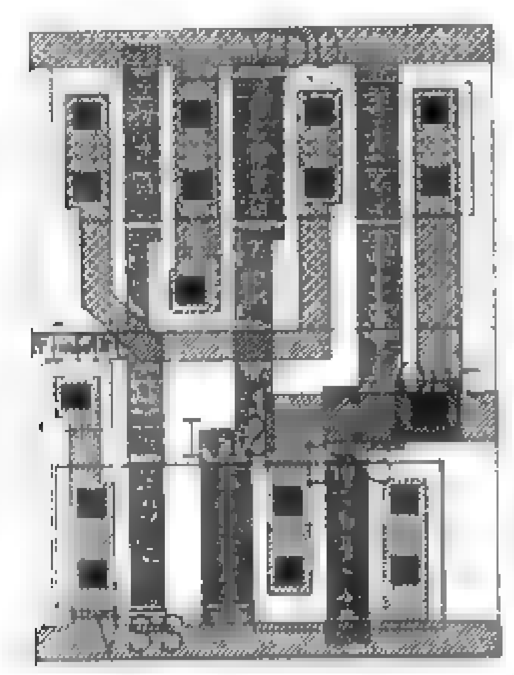


图 5.6　布线格点类型的例子

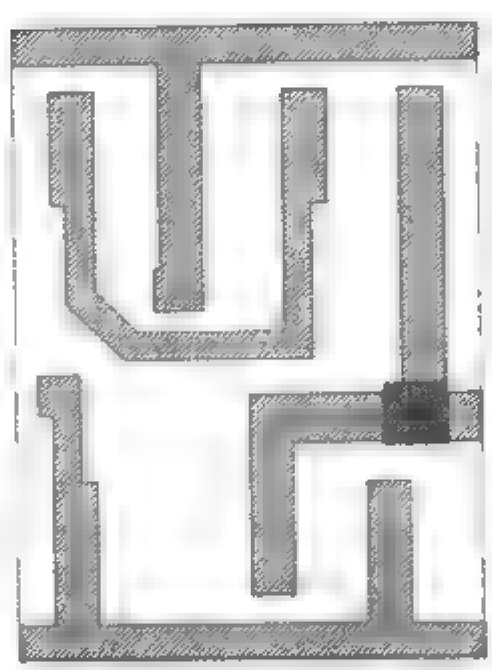
如果布线工具可以使用金属 1 连接单元，那么必须知道什么地方可以走线，什么地方不可以。单元内的布线屏蔽为布线工具提供指定层形状的信息。

图 5.7A 是标准单元及金属 1 面积使用的示例。定义布线屏蔽区域的一个方法就是复制所有金属 1 的形状。

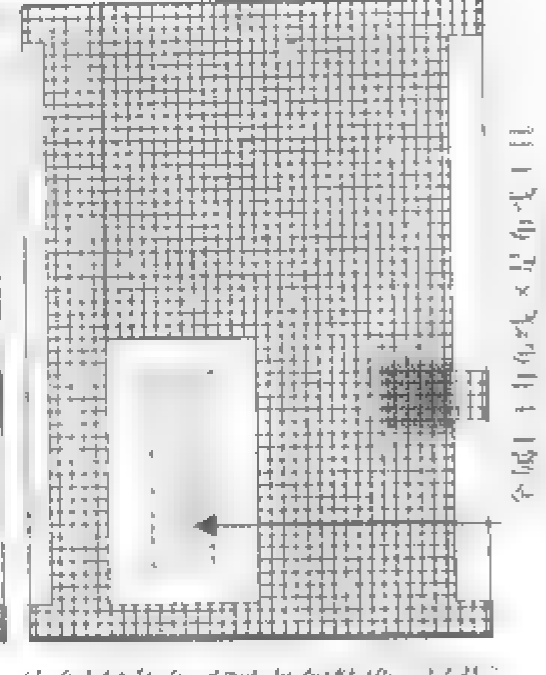
对于小模块（500 单元），布线工具的计算时间和布线屏蔽的定义方式之间有很强关联。布线软件的算法需要为每层建立布线空隙率图，因此屏蔽层的形状越复杂，建立的文件就越大。由于复杂的多边形所要的计算时间更长，也更复杂，布线时间开始呈指数增长。那么怎样让工具占用较少的计算资源，运行更快、更高效呢？图 5.7A 给出了这方面的示例。




单元布局图



给金属1和金属2的“屏蔽”



给金属1和金属2添加的精确“屏蔽”

图 5.7A 单元布线屏蔽的示例

从图 5.7A 可以很容易地看出，巧妙的“屏蔽”对于金属 1 仅有一个多边形，而且，其坐标数据很少。这不仅精确定义了布线屏蔽，也给布线工具在需要打通孔的位置留有足够的空间。

为自动获得有图所示的形状，可以使用简单的宏完成下面几件事：

- 扩大每个布线层上的多边形
- 合并所有的多变形
- 以相同尺寸减小多边形

例如，在决定多边形所要扩大或减小的尺寸时要参考下面的计算公式：

金属 1 宽度 = 0.5 μm
金属 1 间距 = 0.4 μm
版图格点 = 0.05 μm
扩大尺寸值 = 金属 1 间距 +（金属 1 宽度 /2）– 版图格点
= 0.4 μm+0.5 μm/2 – 0.05 μm
= 0.6 μm

图 5.7B 是以上步骤实现的图示。

其目的是获得一个简单图形来覆盖所有的小多边形，这个图形包括了两个图形的间距小于放置其他不相关连接线所需间距的区域。库中的每个单元、每个布线层都采用这种方法，以便大大减少布局布线的时间。

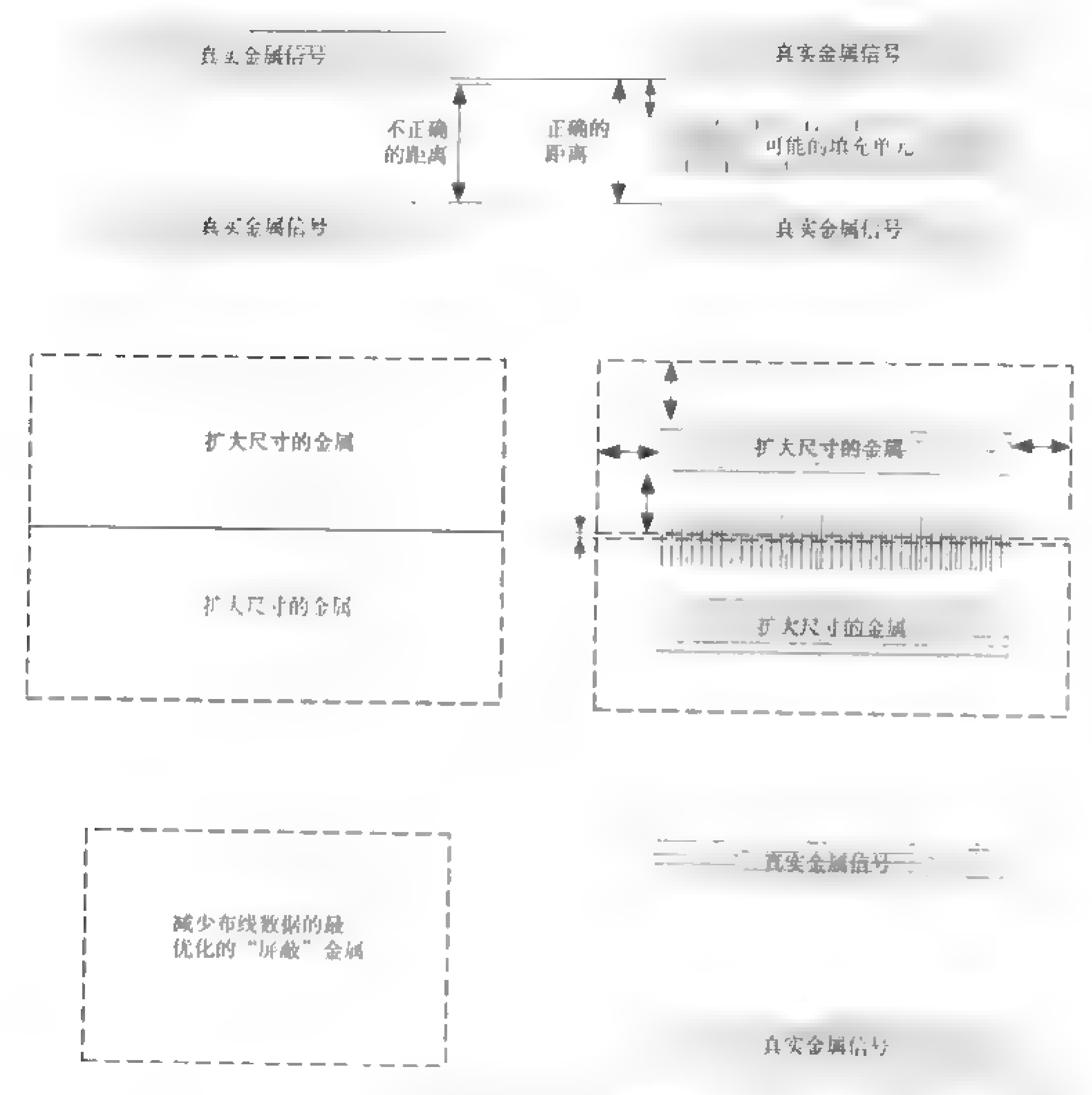


图 5.7B　扩大尺寸的实现图

5.1.5　门阵列

门阵列的设计与标准单元设计有着不同的方法。

不同于标准单元设计的是，在门阵列设计中各个单元的分层不尽相同，而特定库中每个门阵列单元都具有相同的基本层排列。

通常，这些基本层包括形成晶体管必需的层；因此，基本层会包括阱、有源层、注入层和多晶硅栅层。

在此基本层的基础上，单个"标准单元"的功能是通过晶体管互连来组成不同的结构。图5.8就是用来解释如何定义单元的图示。内部的连接层通常是第一层金属或第二层金属。这些定义的层被认为是单元的阴影（shade）和单元之间惟一相互区别的层。

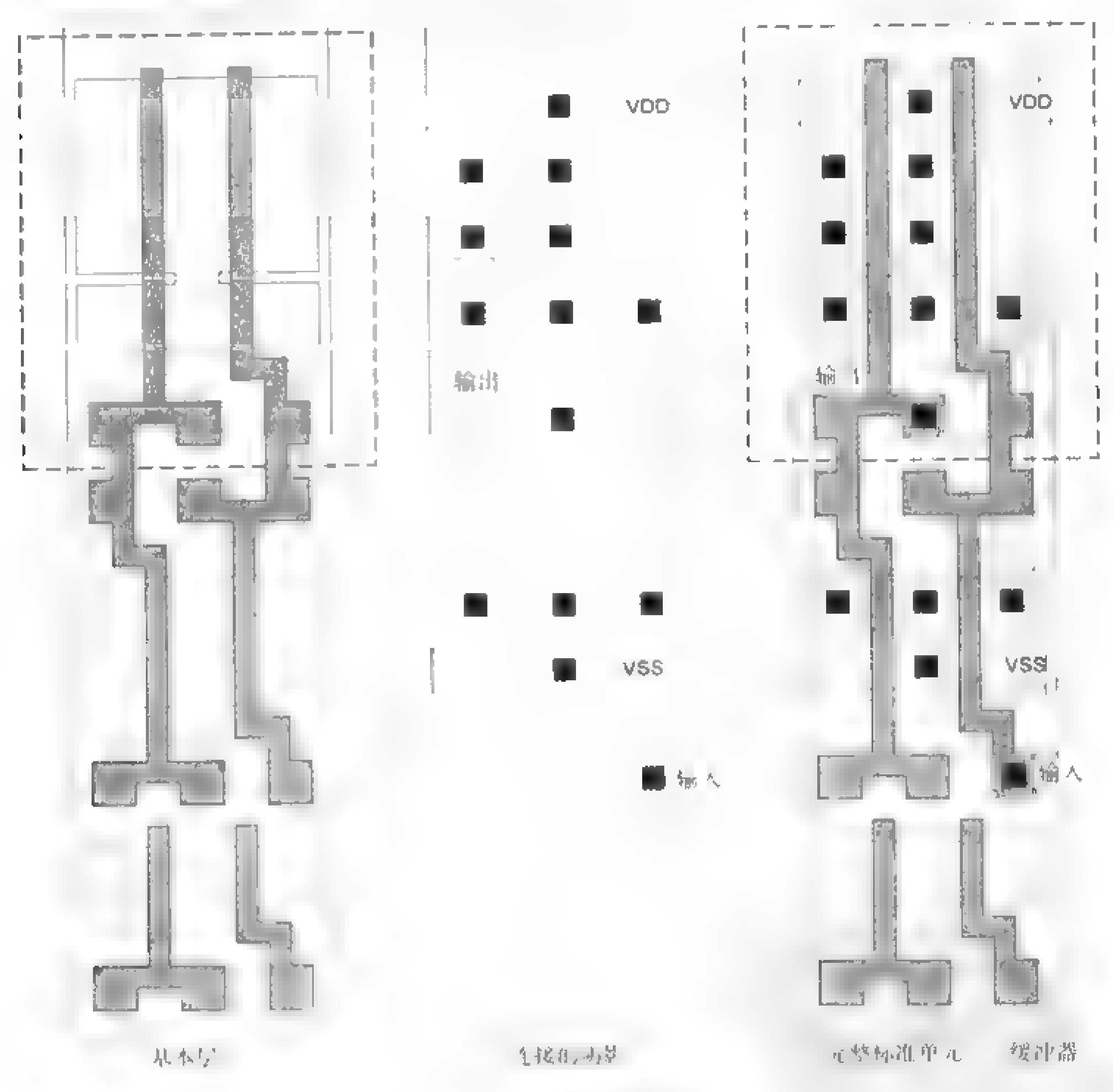


图5.8 门阵列标准单元示例

门阵列这个术语来源于整个设计实现的方式。在门阵列设计中，整个晶圆能够由相同的晶体管阵列或称为基本单元的晶体管组来预制，并用所需单元的阴影部分和自由布线层完成对标准单元及其连接的逻辑定义。

没有任何标准单元定义的基本单元的晶圆通常称为母片，如图5.9所示。需要注意的是，当电路还在设计时，母片就可以提前制造加工了。

表5.1概括比较了门阵列设计和具有完整层次标准单元设计风格的优劣。

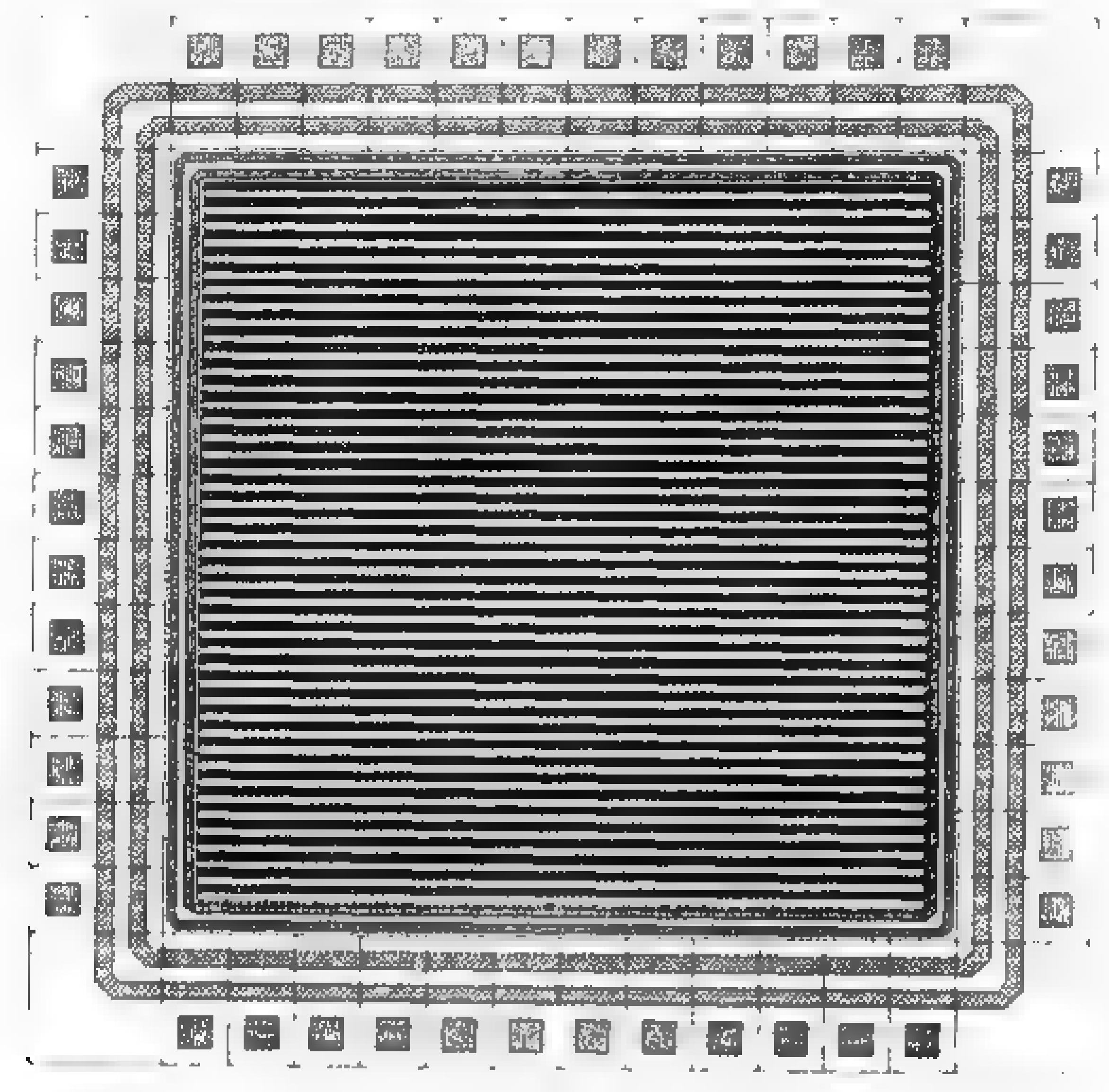


图5.9　门阵列母片示例

表5.1　门阵列设计的主要问题

优点	与完整层次标准单元比较
较快的设计时间	● 预先完成基本层的晶圆
	● 因为放置位置的限制和单元密度的减少，单元的放置和内部连接走线的优化和反复减少了
	● 采用预先制造的晶圆修改已经存在的设计
	● 因为芯片的基本层和芯片的结构都是预先定义的，所以可以预先处理诸如电源布线、门锁（latch-up）保护等很多问题。至少，这些问题的影响比所有层制造完成后的影响其可预测性更强些
性能	● 因为它具有更多的版图优化的机会，所以比可编程IC替换具有更好的性能
增加的可复用性	● 母片可以用于不同的设计或应用
	● 单元设计对于不同的工艺接口更加容易
缺点	**● 与完整层次标准单元比较**
设计灵活性	● 可利用的门的总数固定
	● 版图优化的范围很小
	● 采用全定制模块如存储器的机会很小
单元密度	● 普通的基本层单元设计导致未使用的晶体管的开销

简化门阵列的设计要在预先制造晶圆电路时的高成本和制造工艺的速度优势之间进行平衡。只有制造工艺的最后几个步骤才依赖于设计的应用。因此，门阵列比全定制或标准单元成本更低，也更简单。

为了快速实现设计而预先完成母片的设想是非常有用的。从一系列母片中选择使用合理的结构，处理电路设计尺寸或引脚数目的不同。

另一种体现特殊母片的价值的方案，可能就是单一设计理念在主题上有多种细小的变化。在这种情况下，使用有效的门阵列母片的方法是可行的。

尽管定制模块或"硬核"不能在很多门阵列里通用，但仍然可以被用来完成特殊母片。例如，如果多种应用或设计有共同的需要，如存储器数量相同，那么母片对于节约晶圆面积、降低设计成本就非常有帮助。图5.10是全定制模块应用在门阵列设计中的例子。

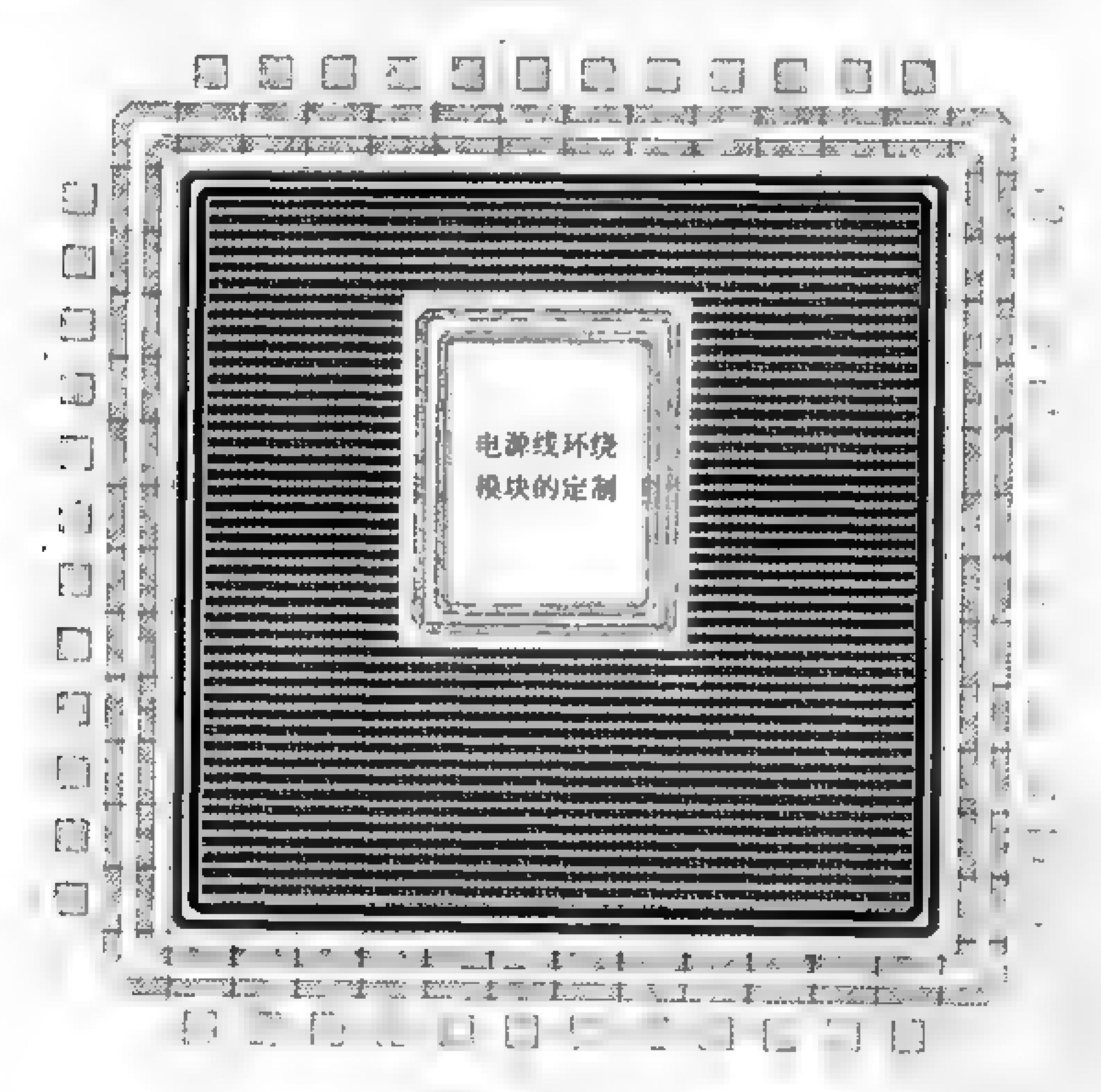


图5.10 门阵列设计中嵌入的定制模块

而反过来，在全定制设计中嵌入门阵列逻辑模块也是有可能的。本书附带的光盘中就描述了一个这方面的例子。

还有什么情况采用门阵列设计风格呢？下面的设计要素证明了它的用途和价值：

- 标准单元设计部分存在风险，也很复杂。
- 标准单元部分的设计落后于进度表。
- 上面的情况都发生时。

考虑到上述情况，特殊的母片的制造和最初设计的实现就要同时进行，所设计的逻辑或标准单元部分就成为了项目的关键。如果采用门阵列设计风格，基本层的制造可以在全部设计结束前就开始。目前基本层的加工时间大概需要一个月，在这期间，设计团队有机会获取和确定最终设计方案。而另一种在完整层次标准单元环境中，可供选择的方案则是在设计定案后整个制造工序将被推迟一个月。

注意本书附带光盘中的门阵列图片是MOSAID很多年以前设计的。一种情况是芯片的几个部分使用特殊定制模块，另一种情况是门阵列逻辑嵌入到全定制设计中，后者是MOSAID为一间位于渥太华的称为Accelerix的公司设计的。

5.2　专用逻辑单元

5.2.1　数据通道库单元

理解数据通道功能最简单的方法是分析一个电路的工作情况，如算术逻辑单元（ALU），它是微处理器的三个重要组成部分之一，其他两个分别是数据寄存器和控制电路。算术逻辑单元在数据的若干位（信号）上进行加、减运算、逻辑运算、屏蔽（masking）和移位（乘法和除法运算），ALU是以数据通道的方式来实现的。

从名称上可以看出，乘法器电路的输出信号是两个输入信号的算术乘积，乘法器电路是数据通道方式实现电路的另一示例。

数据通道单元有哪些不同于标准单元的特性呢?

- **信号的传递：** 与ALU、乘法器一样，电路中通常也存在数据的信号或位的流动。
- **多路信号：** 若干组或总线的信号会同时通过电路。
- **对称性要求：** 信号通过数据通道时，每个信号路径与其他路径的拓扑恒等是需要高度重视的，这能确保不会发生时序的失配，还能根据已知信号进行信号的预测。

如何能系统、有效地处理这些特殊要求呢? 答案是采用数据通道库及相应技术。

我们来分析图5.11给出的电路示例。三个不同的功能块（即F1、F2、F3）处理四个八位信号，得到两组总线输出O1、O2。同时，控制块C1、C2、C3控制功能块的八位数据流。

版图的另一种实现方法是如图5.11所示对电路进行版图规划，但在这种情况下，字节之间信号对称性的要求很难实现。

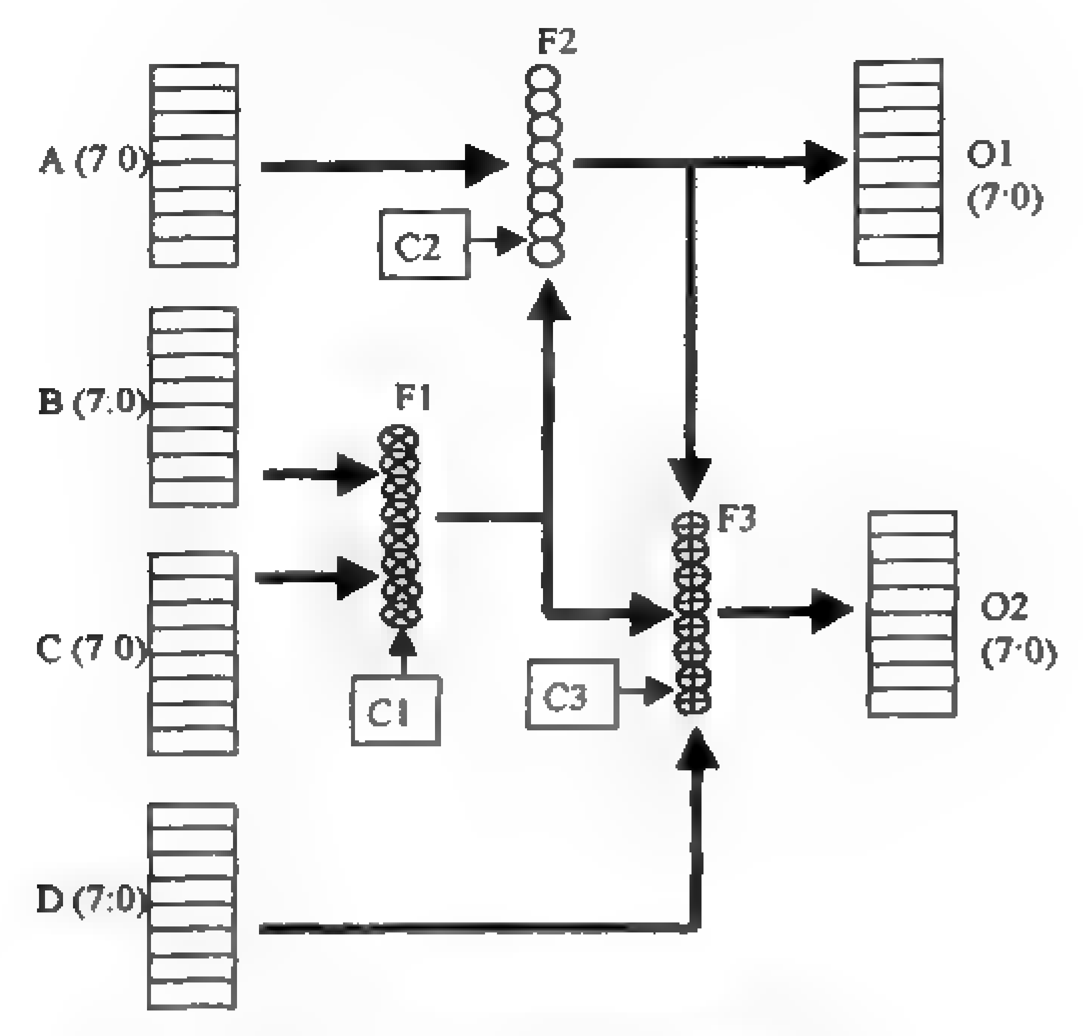


图5.11　数据通道模块的图表示例

这里将应用正确的数据通道技术。

最重要的环节是版图设计一开始就要有整个模块功能的概念。比特流流过信号线的路径、它们的数量和位置、内部单元连通所需垂直路径的数量等都依赖于电路图的连接和性能要求。在这种情况下，设计工程师和版图设计师需要紧密的配合，否则，全定制模块很难达到所有的设计要求。

在正确的数据通道设计中，第一步要考虑一位一位的基本电路，如图5.12所示，特别要注意的是，这一点既应用在电路设计中，也应用在版图设计中。

记住，所有的八位数据流同时受控制信号的控制，下面将介绍八位数据流是如何达到对称的。

1. 把整个功能块划分成小单元，这样F1、F2、F3就成为了独立单元，版图规划中每个功能就能按行实现。
2. 内部版图完成之前定义各个单元的接口电路。它必须包括单元内部和外部的布线要求，还需考虑垂直、水平路径的数目和层，以及任何可能的公共布线通道。
3. 填补单元的内部版图。
4. 完成一位的完整数据通道，设计其他位时就可以直接重复调用已有版图，设计单元应使其在多个位的方向上彼此相邻。

注意，电路图层次和版图层次应该与简单的验证和提取相匹配。所给例子的版图规划与图 5.12 所示的流程相似，它的三个功能块依照所示的顺序放置并以单元的形式实现。

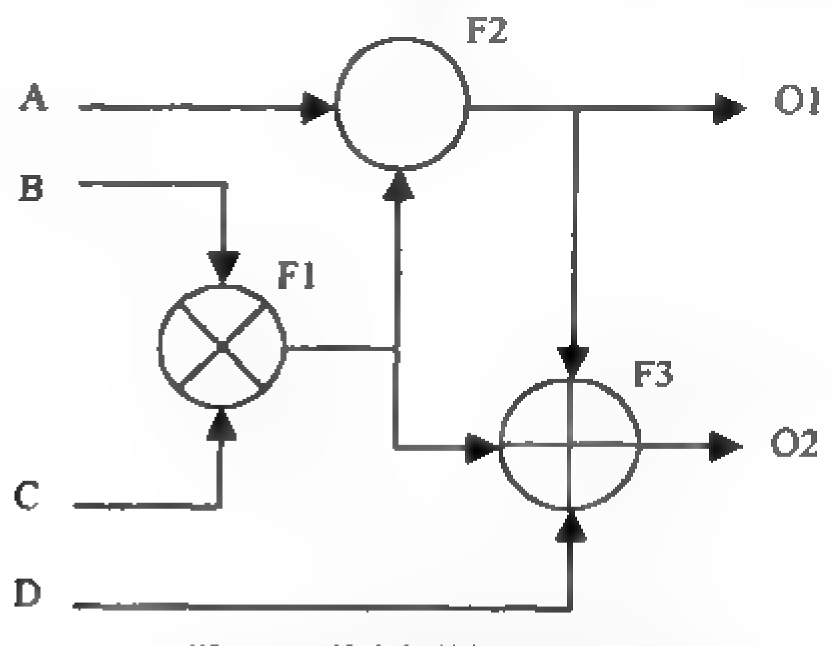


图 5.12　单个位数据流原理图

对于所有位共享的单元，比如控制电路，应该考虑单元分类和版图规划。

图 5.13 是数据通道功能模块一位一个功能的接口设计示例。这是数据通道库单元之一。



图 5.13　数据通道库单元的实例

从图 5.13 中可以看出，数据通道单元有很多值得注意的特点：

- 在单元垂直方向上的接口用地线 VSS 作为起点和终点进行连接。
- 预先确定垂直布线路径作为行（功能块）之间的互连。
- 只要存在预先确定的布线路径，信号在单元的功能块之间传递时就能改变方向。

图 5.14 是完整数据通道电路实现的图示。观察这个基本单元和 3 个功能块与 3 位之间的阵列，可以得到如下信息：

- 如图所示，内部单元布线根据现有的路径进行。信号自由通过，通孔置于线的中心或中心附近的地方。
- 如图所示，连接不同行时有很多线没有使用，专门为单元的内部连接留有足够的备用路径。

- 布线是一位一位进行的，布线单元排列在超出数据通道宽度的地方
- 信号的方向也遵守一定的规则：M1和M3只能水平布线，M2和M4只能垂直布线
- 所有产生控制信号的门都放在功能块各行的末端，布线路径和版图的有效使用面积也以这种方式获得
- 终端单元用于隔离衬底连接和交叠规则（如N阱保护环），逻辑可根据需要放置在两端

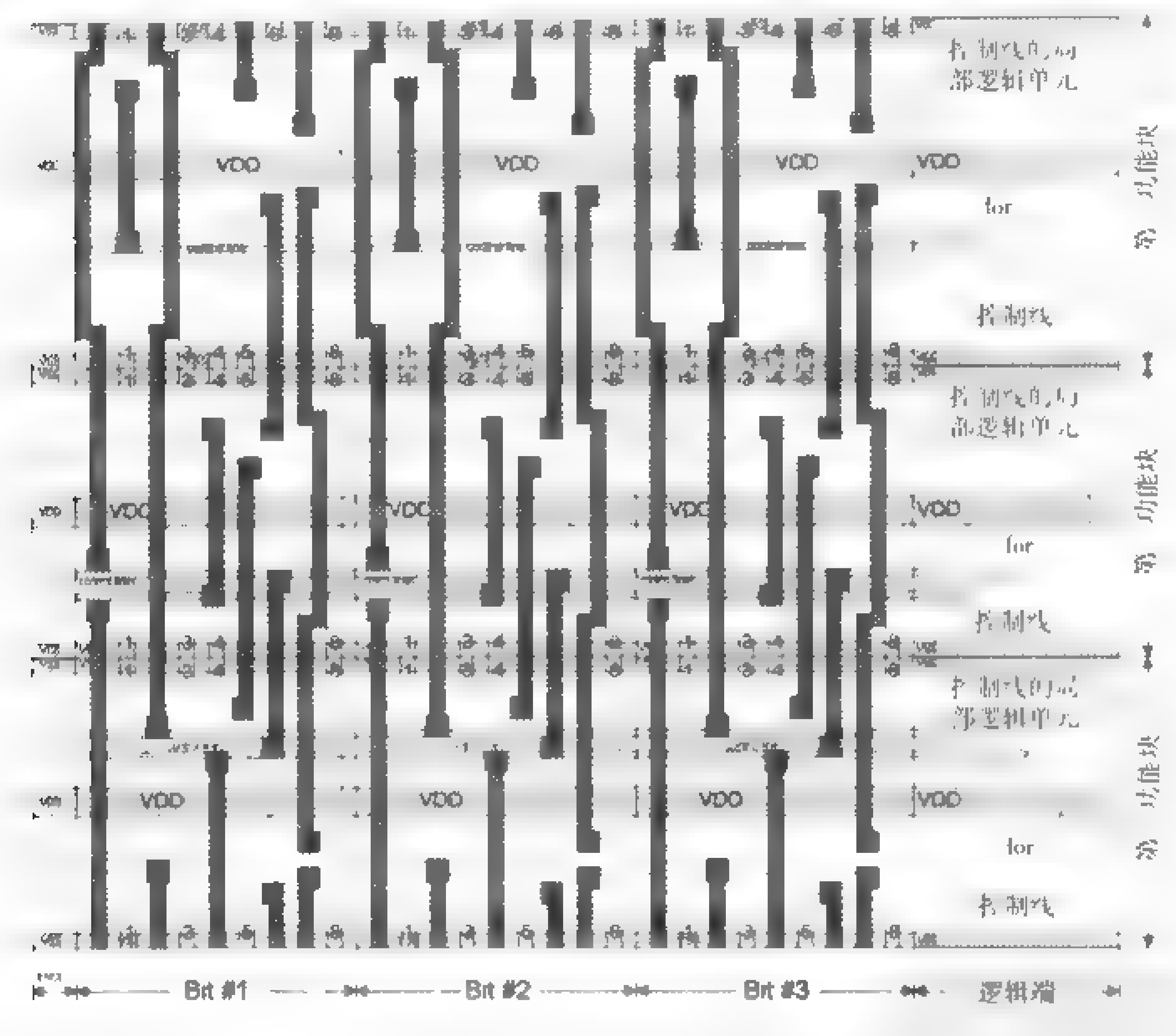


图5.14　三位数据通道示例

在进行单元的版图设计时，理解数据通道和标准单元的库单元之间的差别是非常重要的（见表5.2）

考虑到版图的复杂性，使用本书附带光盘中的彩色图可能会更利于分析

下面是关于采用自动布局布线实现的数据通道模块的一些讨论　有两种自动数据通道方式：一是使用简单的标准库和时序驱动约束条件的P&R标准单元；更高级的数据通道模块方法是使用专门的完全遵循前面所提及的规则的库　数据通道专用布局布线工具专门进行"位"的布局布线，而且对于总线上的信号是完全对称的

表5.2　标准单元和数据通道单元对照表

标准单元	数据通道单元
具有可以和库中其他单元连接的水平接口	利用特定的终端单元，具有仅可以和自己连接的水平接口
具有可以和库中其他单元连接的垂直接口	同上
通常由一个N+和一个P+区组成	不受任何细节的限制和约束。大多数情况下单元布局依赖于晶体管尺寸
假定同一行的单元并不同时工作，计算单元内部的电源线得到平均功耗和电源网格	单元内部的电源线的计算是为了根据已定的位宽度维持规定行的长度，这时总线上所有位的单元同步工作
单元高度固定	单元高度可变，并独立于其他单元
单元宽度可变，依赖于晶体管的数目和连接	布局定义了整个单元的信号"合法"布线路径的数目，并确定了单元宽度
标准化晶体管尺寸	晶体管尺寸可完全利用版图设计定制
单元设计采用最大空隙率（porosity）单元内部的金属2、金属3最小化	单元已经有了预先确定的布线，因此，版图可以利用已知的空闲或已占路径
每个单元只有一个标准功能——如反相器（INV）、触发器（FLIP_PLOP）	设计单元与定义单元上布线一样，有多个功能块（概念与门阵列单元类似）
要提取全部经过布局布线的区域来获得时序信息	要理解数据通道的时序，只需提取一位

5.2.2　时钟产生单元

通常，时钟发生器或者时钟缓冲单元在库中是专用单元，它们专门用于缓冲或是放大负载过重的信号，即系统或芯片的时钟信号。系统时钟的延迟在整个芯片上的分布应尽可能少，因此，时钟分布系统的实现对于版图本身来说就是一个难题。本节将集中讨论缓冲单元。

时钟产生单元的特殊之处在哪里？晶体管尺寸可以很大：独立器件的宽度为1500~2000 μm是很常见的。与之相比，最小尺寸的反相器宽度仅仅为1.5 μm左右。为了有效地实现，这些大晶体管应当使用特殊的版图技术。这种器件值得关注的问题如下：

- 根据电阻和电容情况，优化信号和电源连接。
- 良好的衬底连接——芯片的时钟通常是最高速的信号之一，可能会产生大量的噪声和与衬底的耦合，所以通常这些晶体管用独立的保护环来隔离。
- 减小电源电阻的技术：包括从电源PAD进行的宽金属总线连接，以及大量的通孔。
- 严格遵守电迁移规则。
- 时钟信号的时序特性很关键，因此，版图的提取和仿真是必需的。
- 如果不同的时钟需要同步，它们之间的版图应该是对称的。常见的方法是在不同的位置使用一个可配置的时钟单元。

第7章对更大尺寸的晶体管的版图技术做了详细的讨论，本书附带的光盘中给出了时钟产生单元的示例图片，显而易见，此图片极为复杂以致无法用黑白图片表示。

5.2.3 总线接口单元或滚桶移位器

复杂芯片有很多信号总线,它们使不同模块通过接口连接到外部区域内或外部区域的某一部分,或者连接整个外部区域。然而,许多总线和信号占据芯片中太多宝贵的面积。不同的模块基于不同的时钟工作,因此,解决的方法是给不同模块在不同时间使用的总线增加开关。

总线接口单元,即BIU是一种控制不同时间总线上信号交换的方法。不同信号基本上是不同时间由多路传输到公共总线上的。这些总线带有很重的混合电路负载,减小完全由布线通道消耗的芯片面积是非常必要的。

满足这些要求会遇到以下问题:

- 减小总线电容。
- 减小总线本身和混乱电路所占的面积。
- 实现所有总线连接的对称性和可预估性。

减小信号电容和增大布线面积使用率的方法是直接在信号布线时计划并使用这种专用电路。因总线大量连接产生的寄生负载也用这种方法减小。如果使用公共单元,那么所有总线的连接都是类似的。

就版图设计而言,直接在信号总线下实现版图是最富有挑战性的任务之一。只有在全定制版图方面有经验的设计师才能出色完成。完成这类设计时,版图设计的所有方面都要考虑,例如,面积、尺寸、配置、电容、电阻、对称性和布线所用的层。

本书附带的光盘中给出了这种类型电路的示例,由于它的最终设计非常复杂,以至于黑白图片无法有效地表示。

5.3 PAD单元

每个芯片都有与印刷电路板的外部界面相连接的接口。集成电路是通过它的封装引脚来实现这种连接的。在芯片封装内部,这些引脚连接到金属导线,这部分统称为框架结构。芯片上的最终连接是通过金线从框架结构连接到称为PAD的较大金属区域。

PAD单元是含有大的金属区域或PAD的版图单元。为确保所有PAD在特性上一致,将这些版图结构作为独立单元是很常见的,也是更可靠的。PAD单元由若干结构组成,它们包含以下部分:

- 绑定金属线所需的可靠连接区域。
- ESD保护结构。
- 与内部电路相连的接口。
- 逻辑电路是与PAD功能(如输入或输出缓冲)相关的。

相比较而言，PAD很大，约85 × 85 μm，因为PAD是作为物理机械焊接金线的目标。PAD金属采用最顶层的金属，通常有一个很大的通孔来连接顶层金属和下面的层。

图5.15给出了PAD结构和纵剖图，显示了金属层的组合和钝化或玻璃（overglass）层。注意，钝化层是一个反相层。由于需要，PAD没有被起保护作用的钝化层（保护die的一种玻璃物质）覆盖，因此这个区域容易受灰尘或其他外部物质的损害。还要注意，两个金属层之间由一个很大的通孔连接。至于更多不同通孔的连接，请参阅后面的章节。

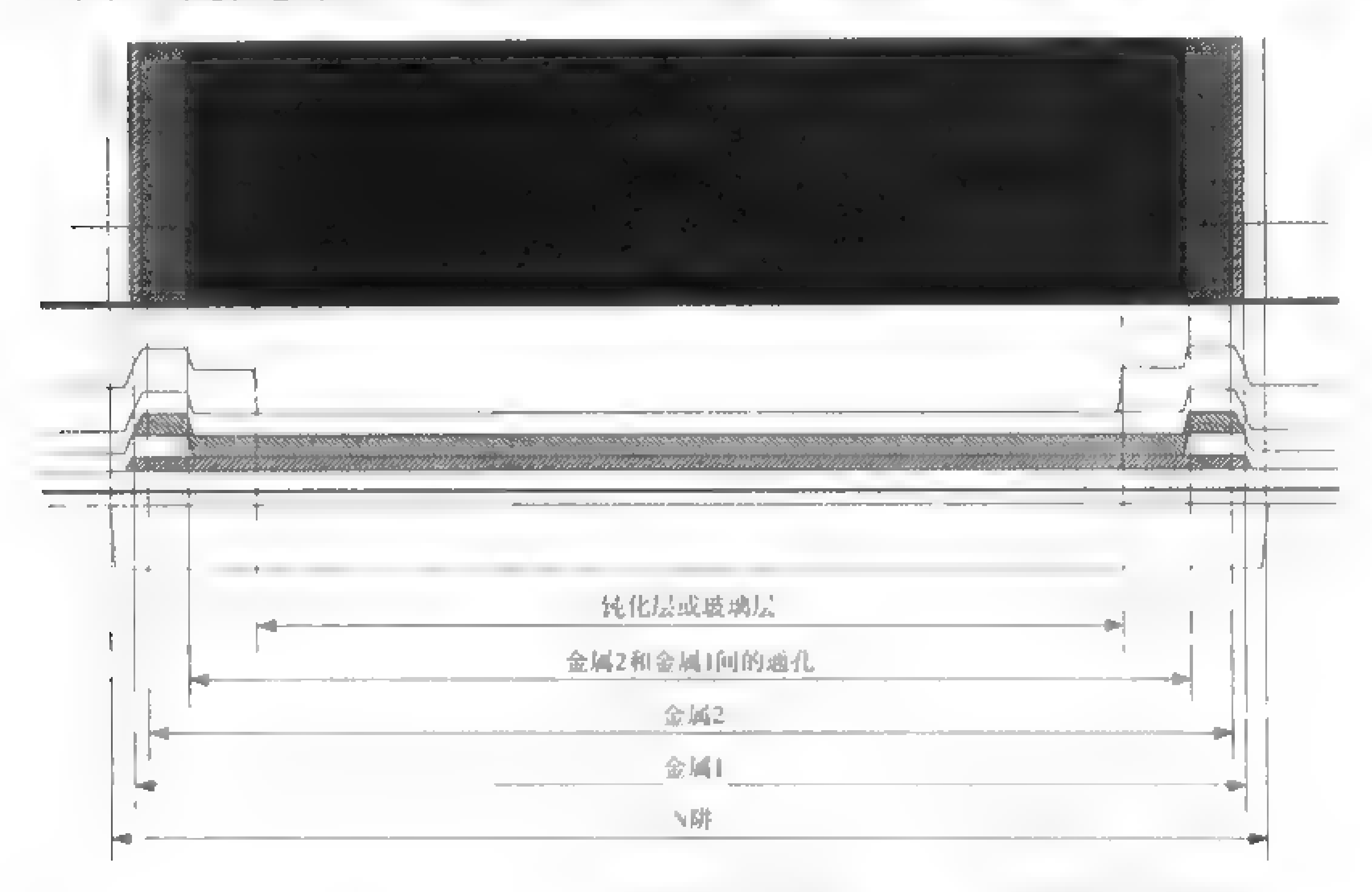


图 5.15　绑定 PAD 版图

注意非多边形区域涉及了一种在PAD周围的可测量的间距，在这一间距内始终存在不可见的电弧。这个特征可以应用于PAD间45°角的走线。

在大多数的ASIC设计中，PAD是成环状的，环绕在芯片的核心逻辑电路的周围放置，所以在版图上PAD单元的设计用的是特殊的地理位置规则。通常，这称为PAD框架（frame）。由于PAD单元有三种典型的类型——电源、输入和输出，PAD框架又称为I/O区域。图5.16和图5.17给出了相关的示例。

图5.16所示的是交错的PAD单元。因为PAD可被更紧凑的放置在一起，所以这种排列能够容纳更多的PAD。在这种情形下，制造商需要一个不同的、更专门化的绑定机器。若能够容纳更多PAD并获得更小、更有效的芯片，则可以补偿绑定的复杂度的提高和成本的增长。由

于PAD必须是一个特定的物理尺寸，在某些情况下，当PAD单元数与逻辑电路数的比率很高时，设计的尺寸可能就会取决于PAD的数量。我们的希望是避免这种情况的发生。

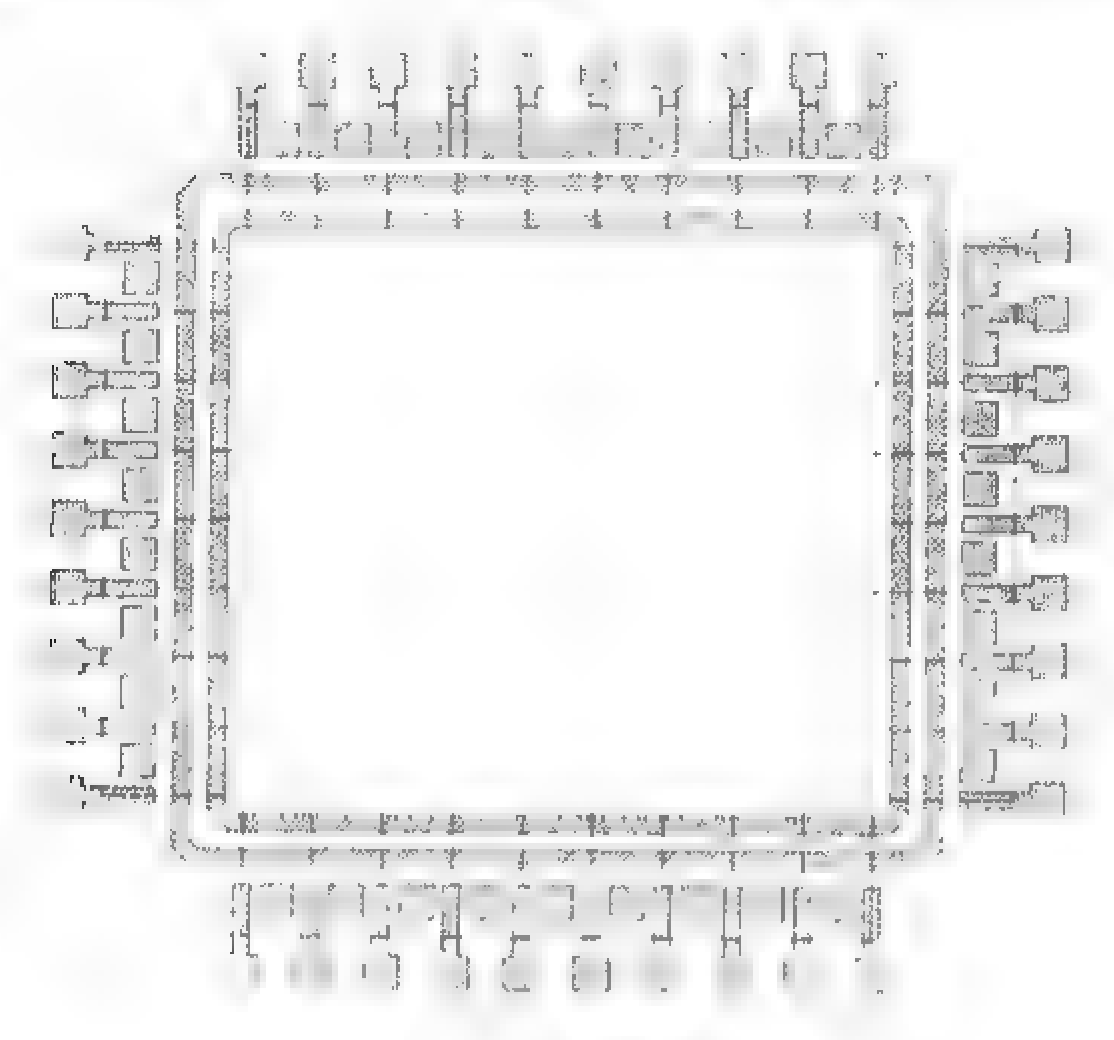


图 5.16 PAD 框架示例

注意，PAD中的金属是排成两行的。供电电源环及与I/O电路相关的晶体管仍然是排成一行的。这是为了电路性能的一致性以及版图设计和集成的简化。

现在来考虑一下PAD单元版图的特殊准则。PAD单元的基本要求就是能将一根耐用的线绑定连接到芯片上。

- **PAD 尺寸：** 要足够大来满足金属线的绑定。这纯粹是绑定机器功能的要求；因此，它被定义为版图设计规则的一部分。
- **PAD 间距：** PAD 不仅需要一个特殊的尺寸，还需要彼此间有足够大的间距作为分隔，避免两个绑定线相互接触而短路。这个规则规定的或许是从PAD中心到中心或从钝化通道的边缘到边缘的间距。此外，设计规则还将定义来自绑定机器约束的数量。这点通过将PAD单元彼此邻接放置来实现，并仍遵守定义PAD单元的宽度的规则。对于双绑定PAD而言，短路不是致命的问题，因为这里的PAD是连接在一起的，双绑定PAD指定的间距是较小的值。

- **PAD 与其他结构的间距：** 为了避免内部电路和绑定线短路，通常在设计规则中定义间距规则。这个规则可以是曼哈顿值，有时指定为由 PAD 的半径定义的区域。
- **PAD 与划片槽的间距：** 为了避免在划片时破坏钝化层或PAD结构，PAD需要放置在离芯片边缘较远的位置。
- **非 PAD 区域：** 有些区域也许不容易绑定，例如芯片拐角处。
- **PAD 处 45° 连接：** 由于PAD单元暴露在外围潜在的较高电流中，因此这里提出了避免尖角和电荷聚积状况出现的规则。简单的示例就是用45° 多边形连接 PAD。
- **PAD 单元原点：** 简化绑定机器提取PAD位置的方法是，将PAD单元的原点设置在绑定的大金属区域的中心。通过简单提取 PAD 单元的版图数据和单元原点，将会产生绑定 PAD 的位置列表。

图5.17给出了其中一些规则的示例。它们是由加工工具决定的，但一般情况下，这些数据有时是可以调整的。设计规则通常在数值上有一定的公差。通过调整这些值可能会出现具有竞争性的优势，因此采用创造性的解决方法并同工艺组沟通来处理这些数值通常是更有成效的。

因为 PAD 框架或 I/O 区域是与外界连接的接口，所以有些特殊要求必须遵守。静电放电（ESD）和有关大器件的问题支配了这个区域的大多数规则。

什么是ESD? 静电放电是大量电荷释放到芯片中，这些电荷对芯片是致命的，因为它可能在物理上破坏接触到这股电荷的晶体管，就像闪电击中了建筑物一样。

ESD的数值变化幅度很宽，但是一个脉冲的持续时间通常很短。ESD 事件会导致结失效、氧化层击穿、有害电荷注入以及内部走线的熔断或开路。

ESD最普遍的来源是人体，当人没有正确地处理IC芯片时，最终的电压将会超过20 000 V。静电对电子元件的破坏从加工到应用每一处都可能发生。在没有控制的、低湿度或拙劣的接地环境下处理元件都会产生损害。

ESD保护结构通常建立在从绑定PAD到内部电路的传导通道中，也是PAD单元的一部分。这些结构的作用如同避雷针，用来把不必要的电荷从内部的敏感电路中移走。

在实现PAD单元结构时，怎样完成或者遵循怎样的设计规则才能将ESD保护的能力最大化呢？在输入和输出结构中采用不同的技术，这将在下面讨论。

5.3.1　输出缓冲器

在 I/O 区域中最复杂的结构是输出缓冲器。这里我们将会遇到很多有关ESD的规则。

输出缓冲器是个大驱动器，它将信号输出芯片。这些器件的宽度范围为400~1000 μm。其尺寸取决于自身的频率、功率、电压电平、电流驱动和功能等。这些大的晶体管必须仔细布局，因为它们需求的区域是高度敏感的，直接影响着芯片的尺寸。

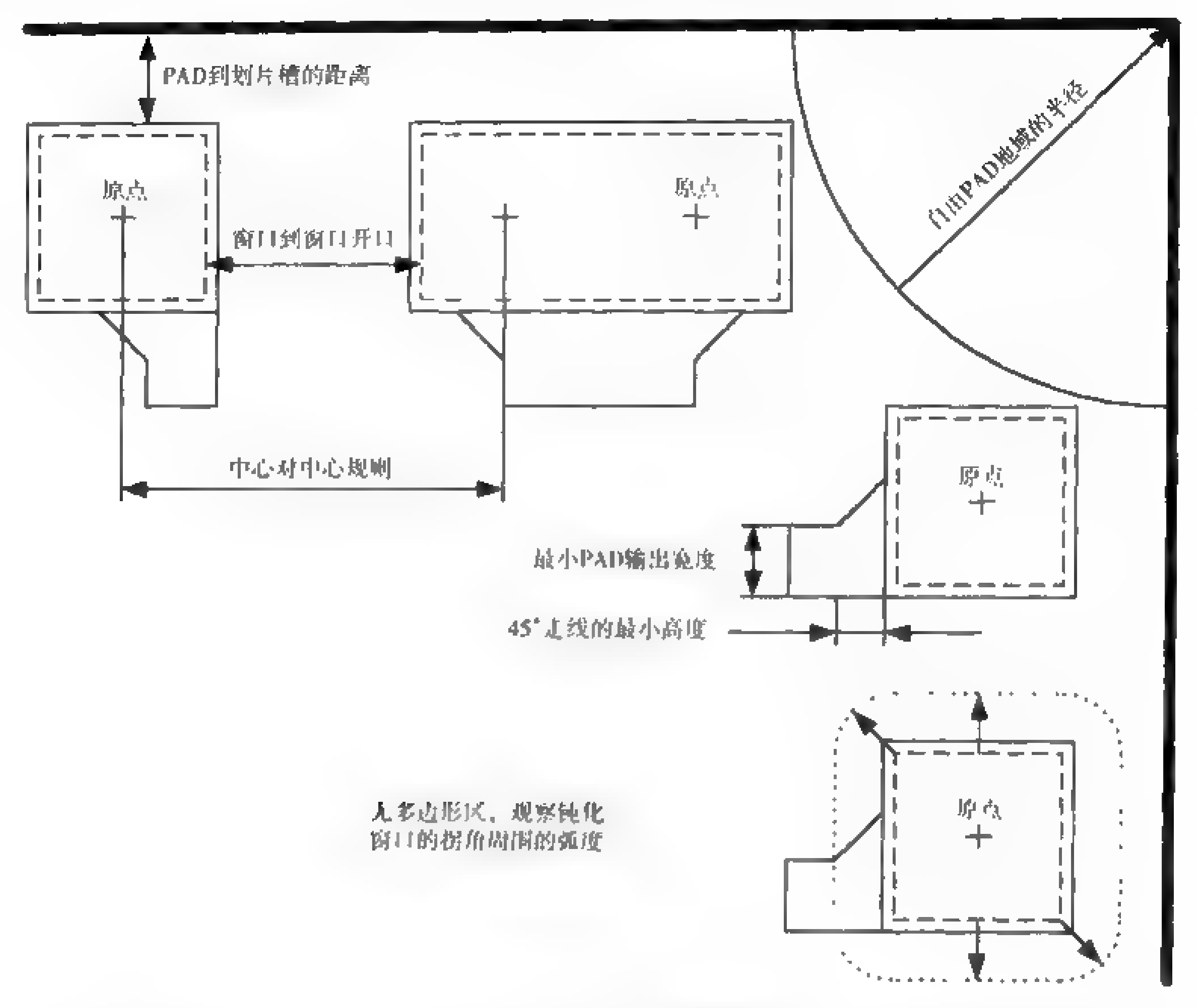


图5.17　PAD单元设计规则的示例

虽然不能说明输出缓冲器设计的所有特征，但是这里将阐述一些普遍的问题和解决方法。在图5.18中，我们将解释几条特定的输出缓冲器规则和指导原则：

(a) 只有信号通道通过接触孔或通孔换层时，金属、多晶或是有源区的每一层才都会被制成45°的多边形。其原因是通过避免拐角就可以避免电源的起伏（surge）和在90°拐角周围电荷的积累。

(b) PAD上部分接触孔和栅的间距比正常晶体管的大得多。原因是这样能够增加从接触孔到多晶栅的有源区电阻并减少通过栅的压降，从而提高晶体管的ESD保护特性。同样，有些情况下，为了增加这些区域的源区电阻，也使用了附加注入层。

(c) 电源部分的接触孔和栅之间的间距可以是最小的或者和(b)中的一样大，取决于工艺要求。需要注意的是，电源引脚很容易受ESD击穿而遭到破坏；因此电源上的部分PAD也需要设计为相同的ESD保护结构，这是很重要的。

(d) 如图5.19所示，在源和漏之间的电流有不同的均匀电场弧。当接触孔在晶体管中间时，电场线是等同且均匀的。

当接触孔安置在源/漏的末端时，电场线聚集在有源区的边缘附近。在这个区域通过栅的电阻会变得较小。如果有ESD击穿，电流将会选用最小电阻的通道并在这一区域的边缘附近趋于集中。这里更容易产生从源和漏到衬底的"击穿"。

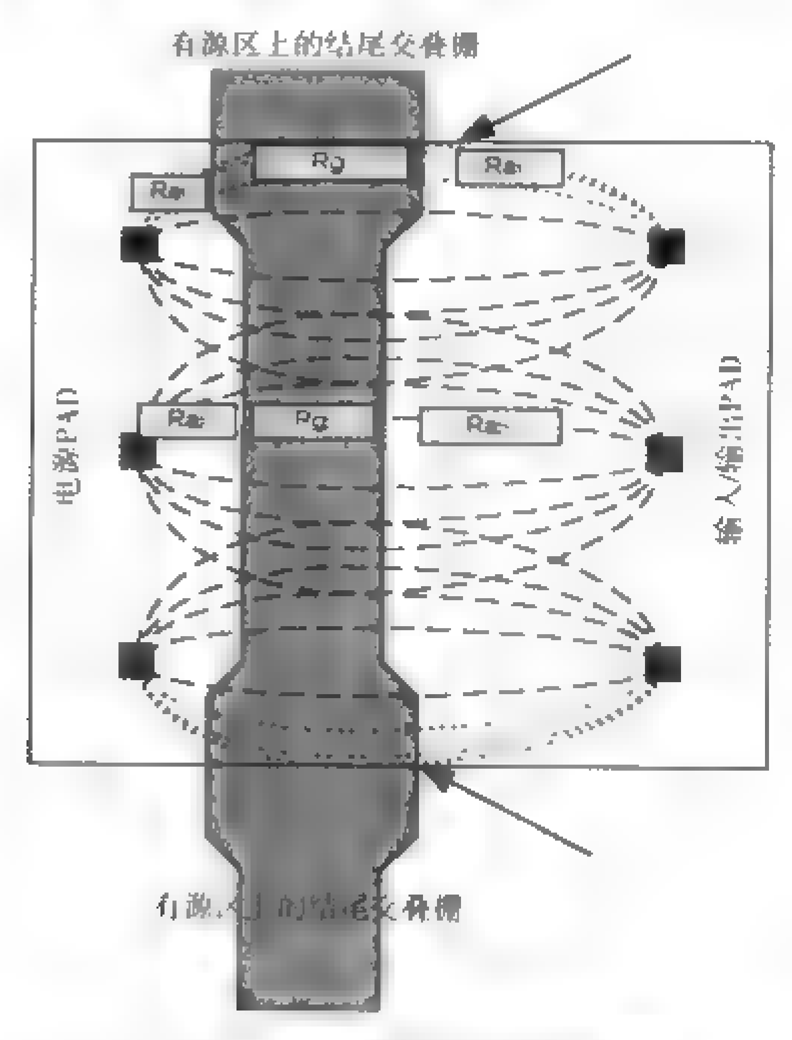


图 5.18　输出缓冲器终端交叠规则

为了避免晶体管边缘的电流聚积，我们通过扩大栅的长度来增加这个区域的电阻。增加的栅的长度应该使等式$Ra1 + Rg1 + Ra11 = Ra2 + Rg2 + Ra21$成立，同时使ESD击穿的电流沿着晶体管的宽度均匀分布。

实际的现象更加复杂，但对于版图的用途而言，希望等价电阻的示例能足够好。

图5.19给出了栅交叠端的两种选择，这在图的顶部和底部可见。底部的实例则演示了将多晶与另一层相连的情况。

(e) 规则(d)中的部分包括特定于有源层边缘的其他相关工艺规则。每个芯片加工公司都有ESD方面的专家，所以我们建议在完成设计之前向专家了解一下相关的要求。

(f) PAD输出连接宽度的建立有很多方面的考虑：如电子迁移、金属的电阻、封装连接的阻抗和感应系数、输出晶体管器件间的等效负载及其他问题。在给PAD连接选择合适

的宽度时，有大量的近似和“艺术”。通常，工艺组为I/O区域的所有器件提出了很多典型的“经过验证或被推荐的”值。

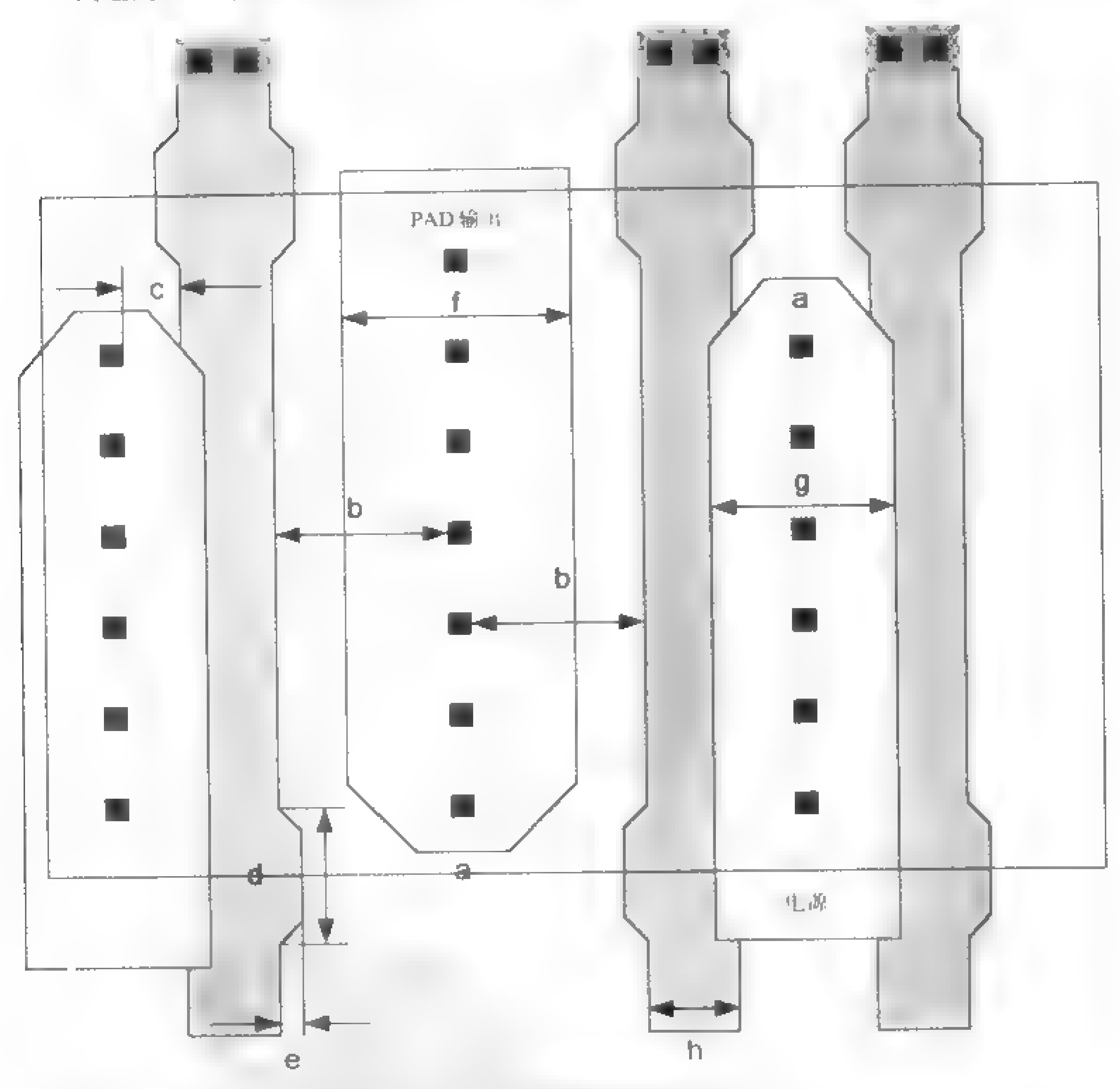


图5.19　输出缓冲器晶体管设计

(g) 总体上，电源连接的宽度通常取决于输出晶体管的功耗和减小晶体管的电源电阻。在示例中，所有的连接都在金属1。对于目前的4~6层金属层工艺条件，输出缓冲器的电源线通常采用金属3或金属4。

对于需要同时进行32位或64位数据转换的缓冲器芯片而言，有大的芯片功率需求。注意，一般输出缓冲器是由特定的或隔离的电源线提供电源的，这些电源线没有连接到其

他晶体管上，而是直接连接到独立的电源PAD上。不同的电源也许在引线框架或封装的外面连接在一起，但是在芯片上，输出缓冲器是连接到VDDQ或VSSQ上的。

(h) 还要注意的是栅的长度。通常横跨在栅上的电压是高电压，所以栅的长度比一般的要大一些。此外，可以从工艺人员那里得到帮助。在不同电压上测试不同的栅的长度，并基于实验和所设计芯片的性能、功率及ESD要求推荐合适的栅长度。

最后需要注意的是：当处理特殊版图单元如输出缓冲器时，给这种区域列个清单并仔细审核是明智之举。

5.3.2　输入缓冲器

输入缓冲器从芯片的外界接收数据。在I/O单元的特殊情况中，许多ESD保护结构和技术都应用在输出晶体管中。

然而，输入保护结构是用来保护弱晶体管的，这些晶体管用来缓冲给内部使用的外部数据。这些器件由工艺人员设计和测试，如果生产者能提供特殊的工艺器件，则这个器件已经过流片验证。

很多时候，在工艺验证完成以前，需要进行输入缓冲器设计，因此，设计输入ESD保护器件是很棘手的问题。图5.20给出了一种方法的示例。

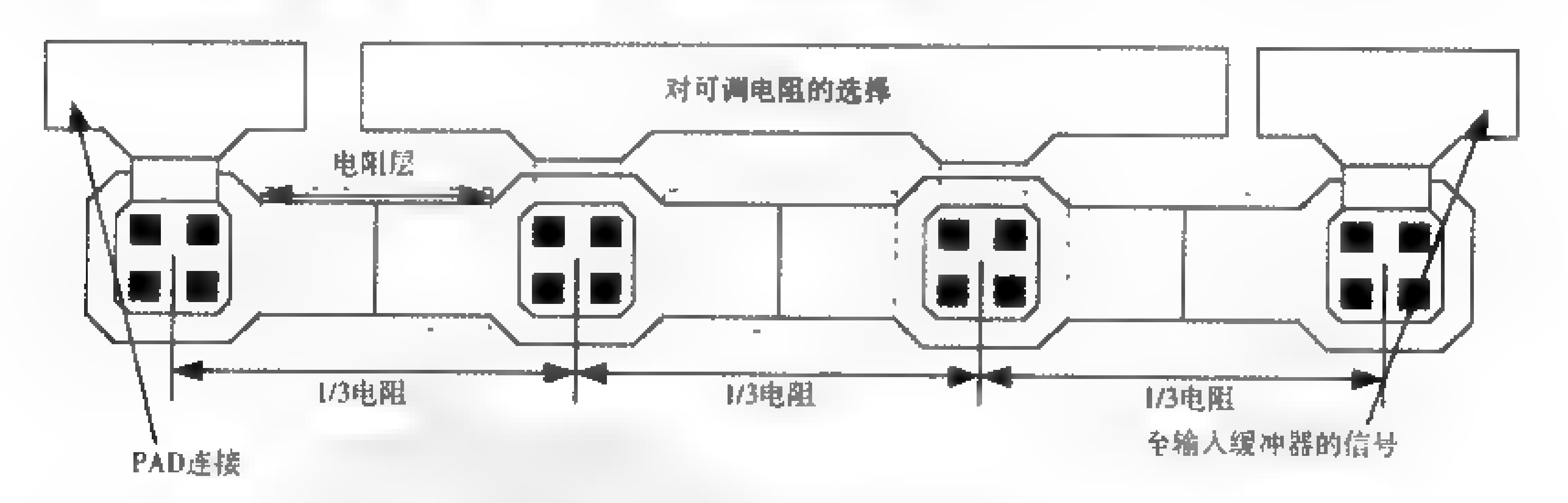


图5.20　输入ESD保护电阻

解决方法是使用一个简单的可调节电阻器。注意，电阻器是由45°拐角的有源多边形组成的。通常，有源多边形是由特殊的ESD注入层包围，这个注入层有着提高电阻器精度的作用。

这个电阻器被分成三个等同的区域，目的是方便以后的调节。这种调节仅仅用于原型阶段，产品设计会选用一个固定配置。在样品评估期间，可以利用聚焦离子束（FIB）仪器来调节电阻，这种仪器可以添加或者切断金属线，还可以添设旁路以避免损坏晶体管。

关于电阻器的设计有许多因素需要考虑：

- 电阻器宽度和长度的选择是基于ESD要求和功能特性的。
- 建议使用一个以上的接触孔的信号通道连接可能的大电流信号。
- 金属线的宽度不应该取最小值，并且必须符合某些电子迁移准则。
- 金属接触孔之间的虚线代表有效电阻的主要区域。它不仅是有效电阻的可用区域，也可作为LVS产生器件的绘图层。LVS器件将确保不会因为疏忽使单元发生改变。

5.4 存储器设计叶单元

存储器版图设计对于初学者而言确实具有挑战性。存储器的相关版图单元的设计会涉及到制造工艺、电路构造及特性问题等方面的详细知识。

存储器的类型很多，如SRAM（静态存储器），但在众多的设计类型中最富挑战性的是动态随机存取存储器，又称为DRAM。图5.21是DRAM芯片运行模块的版图规划图。

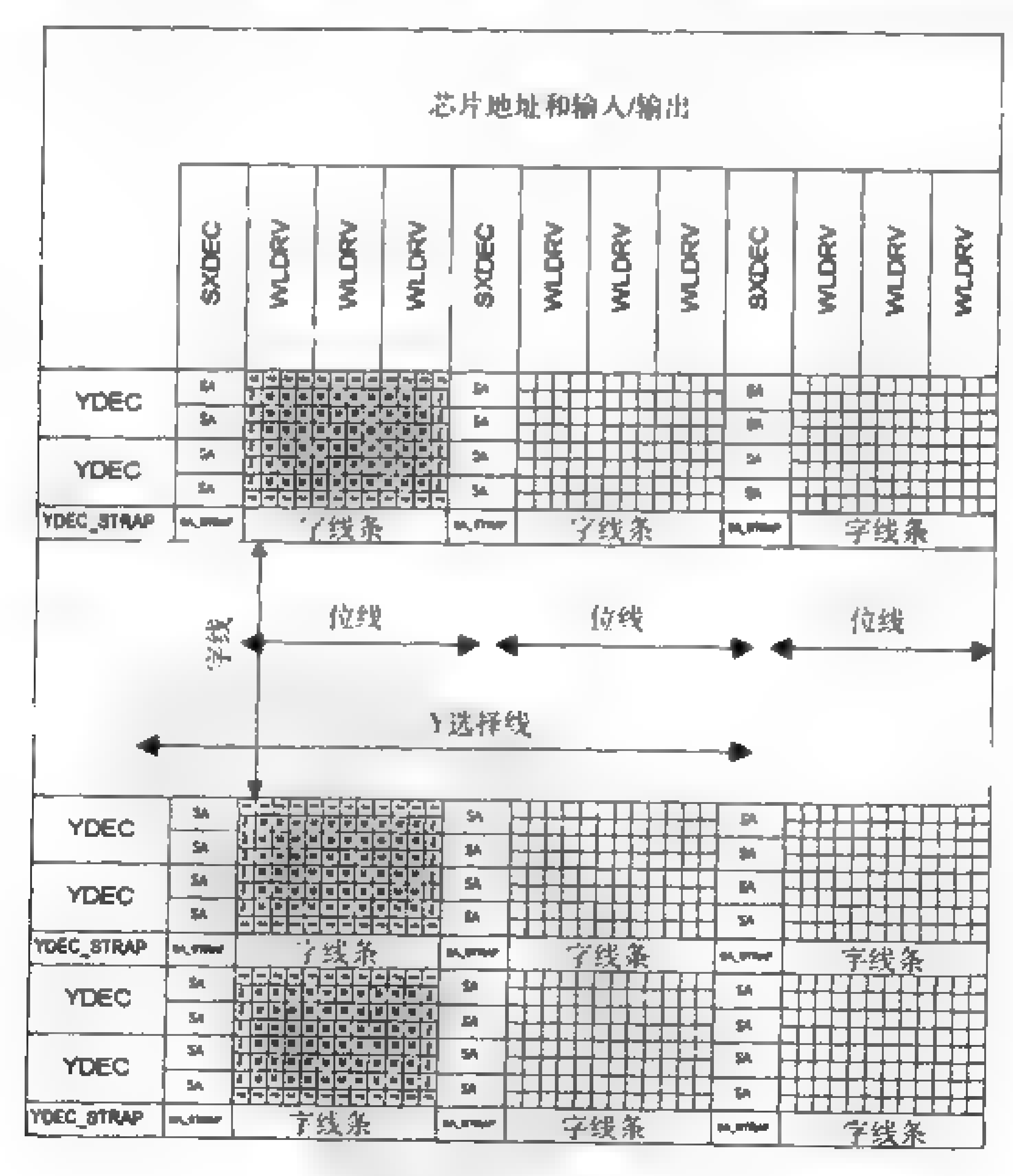


图5.21 DRAM结构

首先让我们看看电路中的基本存储单元是什么样的（见图 5.22）。这是因为基本存储单元的制造工艺是相当复杂的，如图 5.23 所示。

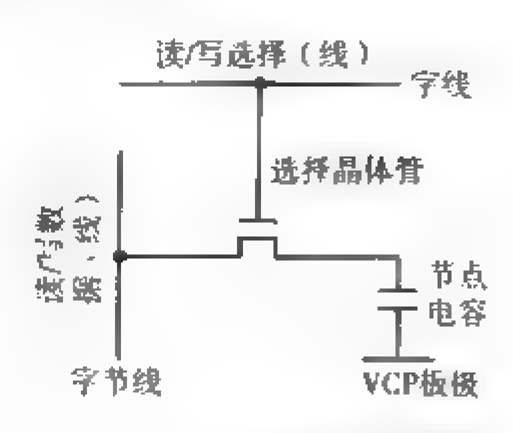


图 5.22　DRAM 单元原理图

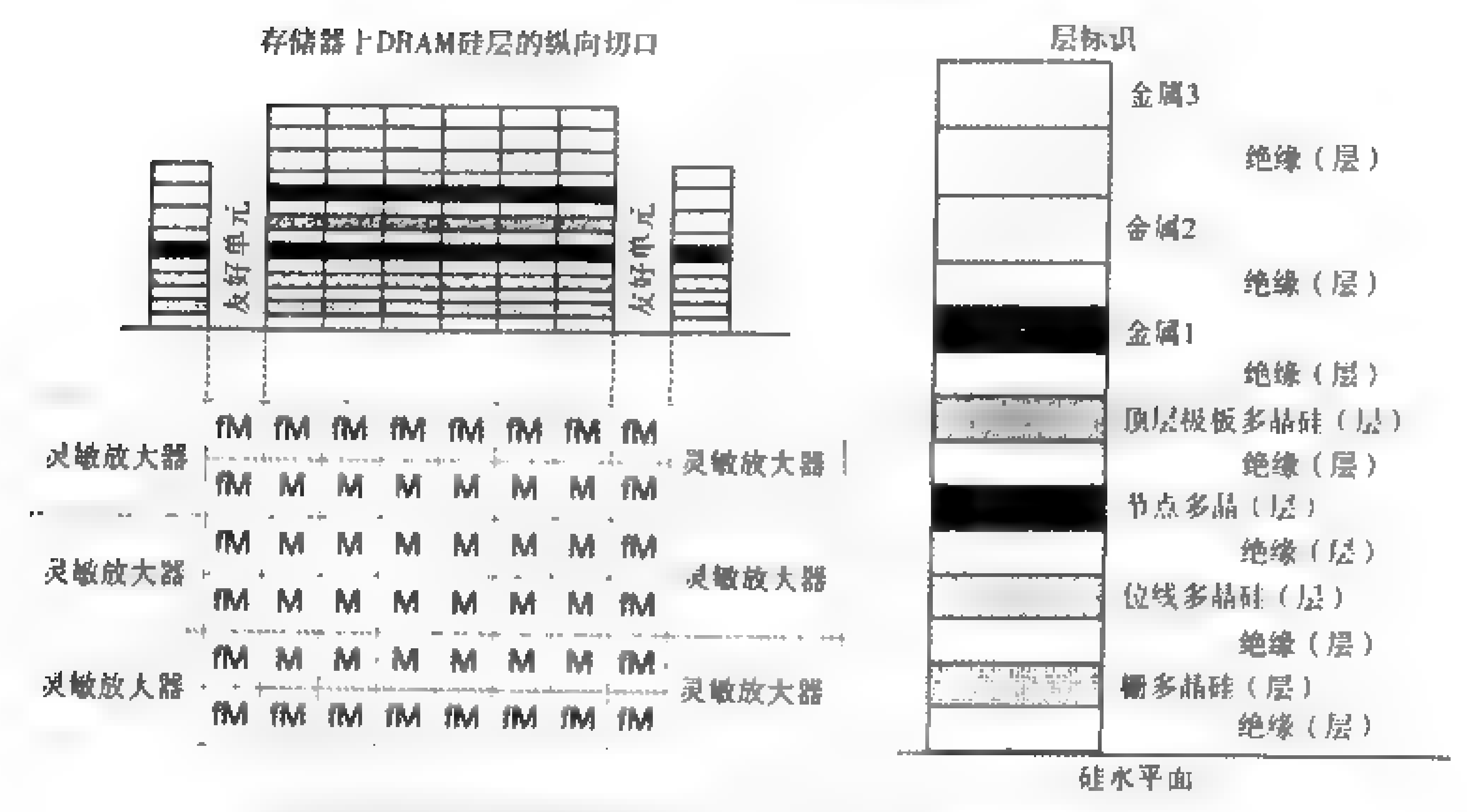


图 5.23　典型堆栈电容式 DRAM 工艺的纵剖图

堆栈电容式 DRAM 存储单元有特殊的层，这些层能够形成存储单元电容器。通常，任何 DRAM 存储单元的版图都采用特殊的制造工艺，而且都是公司的机密。

如图 5.23 所示，在制造工艺方面，DRAM 存储单元的位置是很高的。通常节点和平板多晶层（plate poly layers）只允许在存储单元区域使用。因此，灵敏放大器最多只用两层多晶，且此区域布图影响并不大。"友好"（friendly）单元是用来连接存储单元的规则模块和存储单元阵列外的不规则模块，而这个单元的垂直布局则因阵列外的模块面积的尺寸变化而减小。"友好"单元连接各层也给稳定信号（如电源）提供了保障。

如我们在第 4 章中提到的，存储器的设计对版图有很强的依赖性，事实上正是如此，存储器的版图是最先完成的，接着才是原理图的设计。这样做的原因很简单：存储器单元会被重复

使用很多次，所以减小尺寸才是至关重要的。最后在此基础上我们再完成存储器单元阵列周围的电路。这使得版图的设计要满足间距受限的（pitch-limited）版图设计要求。

间距受限的版图是版图设计中的一种，在这里所考虑的单元版图在某一维上是受到限制的，并且必须与“决定单元”（leader）相联系。大多数情况下，“决定单元”就像存储阵列中的存储单元一样，也是个重复单元。因此，版图规划的重点应集中在减小不受限的维的尺寸上。

图5.24是展示存储器芯片的示例。在存储器芯片中WL（Word-line，字线）驱动器、字线条（strap）以及灵敏放大器单元都是间距受限的版图设计的例子。

图5.24 存储器阵列中的间距受限单元

分析图5.24所示的存储器阵列，可以发现单元间的间距和存储单元内的间距是不同的。例如，字线驱动器具有两个单元的间距。

单元间间距的匹配使得这类版图设计富有挑战性。在任何情况下，当两个单元间的电路出现失配时，后面的单元将不可避免地要做多方面的修改，以符合“决定单元”的要求。

通常，决定单元经过高度优化，用以实现满足最小设计规则的专用电路，这使得单元的尺寸受一系列具体规则的限制。

在确定了详细的规则后，通常很难使决定单元周围的间距受限单元满足统一的间距要求。后面的单元将会有不同的电路要求，而这些要求可能会与限制决定单元的关键设计规则不匹配。

在存储器单元的例子中，设计规则与间距受限电路间存在差异是很普遍的，因为间距受限的布局不像存储器单元的布局那样有规则。这些差异使得版图设计变得更具有挑战性。

下面将会讨论间距受限版图单元的例子来阐述前面的概念。在所有情况中，存储器单元的间距限制这些单元在这一方向的尺寸。

5.4.1　字线条单元

在间距受限的版图概念中，由于多晶硅和金属字线之间的接触孔，字线条单元的布局与决定单元的布局不同。存储器单元是建立在没有接触孔的金属间距上的，这种间距相当小；而接触孔是加在字线条尖端上的（the strapping point）。因此，在字线的方向上只有很小的空间可以打接触孔。

为什么字线条单元是必需的呢？字线条单元是一个值得去认识理解的单元。它是纯粹用来解决电路设计中出现问题的版图方案，而这个方案可以通过很多种途径来实现。

字线驱动器通常具有由 1024 个存储单元组成的门级负载。这是一个大容量的负载，并且较长多晶栅产生的电阻使得字线的延迟受到抑制。金属线的电阻率通常比多晶栅的低三到四个数量级，所以将字线捆扎（strap）成规则的条以减小字线延迟具有重要的意义。

字线条的使用频率和布局主要取决于多晶栅的电阻率、字线特性要求、面积、金属和接触孔的电阻、字线间的对称性以及失败分析考虑。图 5.25 是不同的字线条（包括其正面的和反面的）示意图。

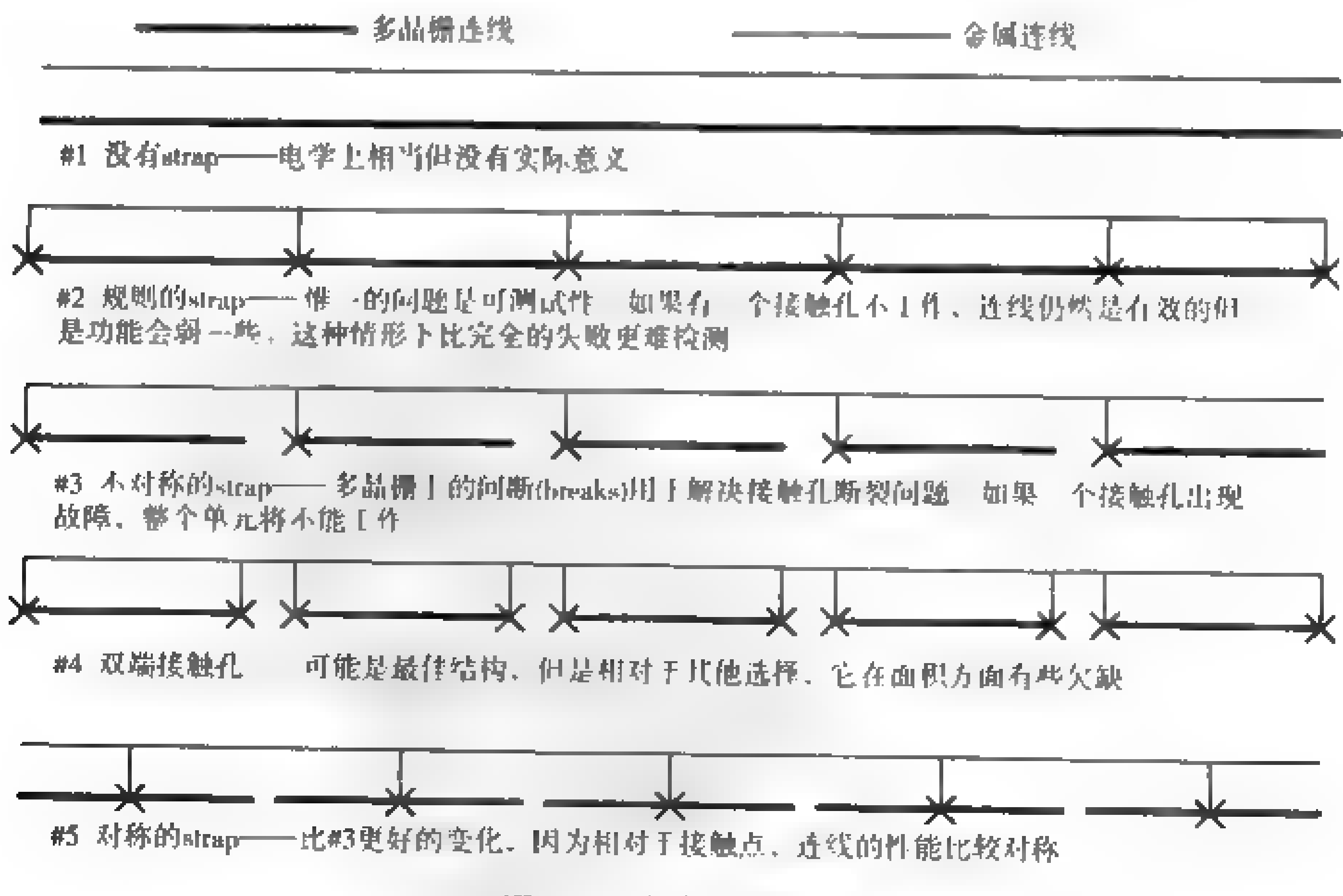


图 5.25　字线条示意图

示意图中假定字线条的频率是已知的。虽然字线条的频率主要取决于字线的特性要求；然而，还要考虑面积的影响。

在设计字线条版图时，需要让字线以最小间距穿过没有接触孔的存储单元，以使每一条连线都打上接触孔。

解决办法是交错连接接触孔单元，分析图5.26，我们就可以明白如何得到总线连接所需要的宽度。

核对本书附带光盘中的字线条的彩色版本示意图

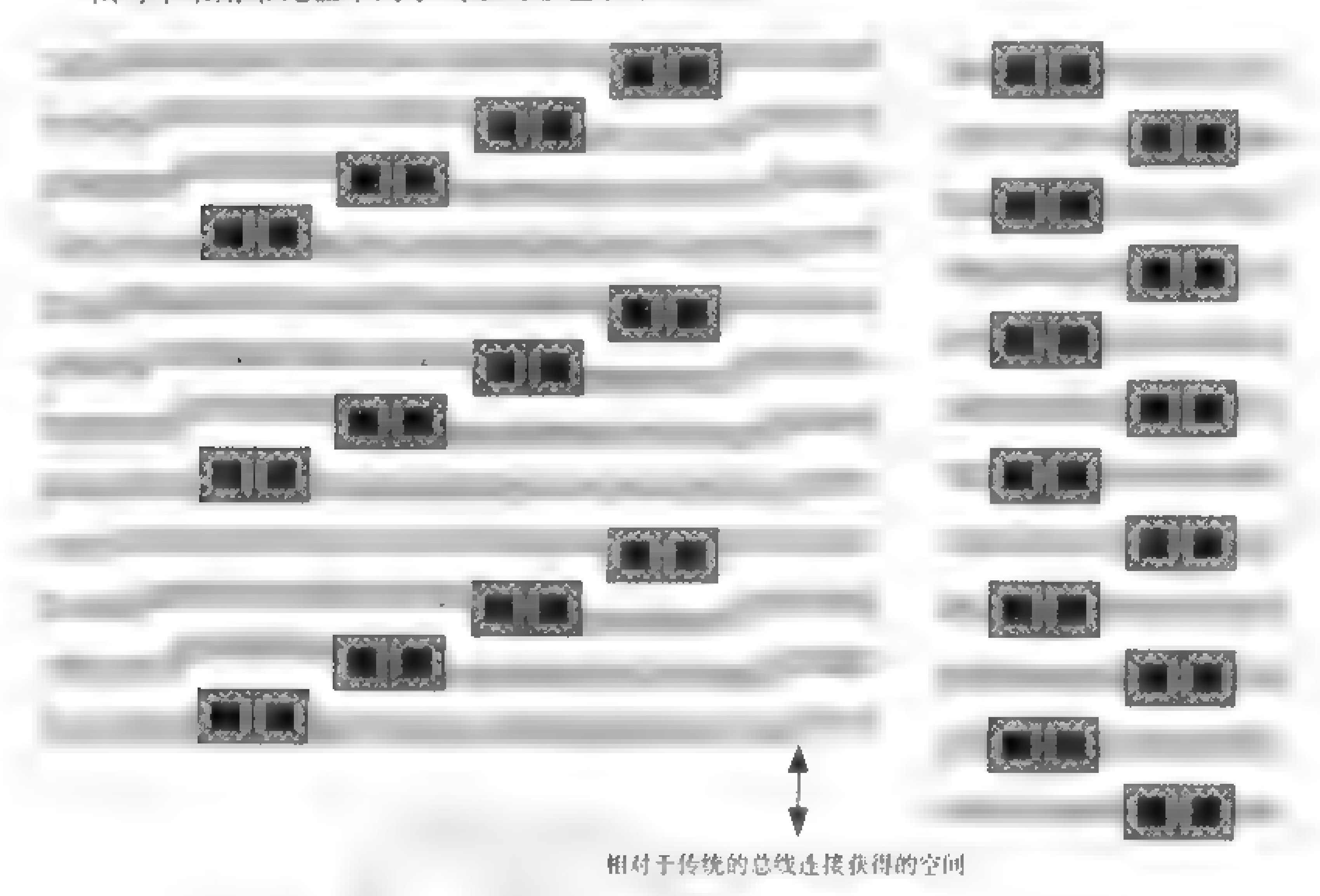


图5.26　连线strap版图的实例

5.4.2　字线驱动器

与字线条相比，字线驱动器是一个更复杂的单元，它不是在字线间距中打单个的接触孔，而是需要实现整个驱动，比如，用经过特定优化的CMOS反相器或二选一器件来减小面积并满足特性要求。

虽然这种工艺的详细介绍超出了本书的范围，但本书中涉及到了这种版图的基本方法

除交错互连的晶体管以外，驱动器和字线条一样采用了必需的交错连接，字线驱动器环境下如何确定晶体管的最小间距也是有很多种方法的。

这里有另外一些注意事项：

- 因为没有和其他单元直接相连，所以需要建立类似于数据通道单元方式的单元。把字线方向看做是数据通道的方向。用垂直于字线的方向作为控制线方向。
- 如果制造工艺不成熟或正在实现新的电路，大多数的工艺工程师在碰到间距受限版图设计时都会对一些设计规则进行协调。通常重大的面积冲突问题可以避免，由于驱动器都是规则模块，且都在工艺很容易控制的存储单元阵列附近，使得设计规则方面的要求就比较宽松。
- 通常 DRAM 存储器有多组电源如高压电源 VPP、衬底负电压电源 VBB，因此，除了完成所需电路外，还应处理和规划这些阱和连接。
- 交错连接的电路是导致不同字线驱动器性能不对称的根本原因，因此，将这种差异最小化并在最终实现验证中模拟出来是设计的目标。

现在可以设想，如果试着在存储单元的间距中交错连接驱动器单元，那么字线驱动器上没有限定的尺寸将会快速增长。图 5.27 是一个可能实现的布局结构。

5.5　激光熔丝单元

因为不同类型的IC设计如微处理器、图像加速器、ASIC等的复杂性，芯片确保没有错误之前要完成大量的仿真工作。问题是在多数情况下，市场方面的压力使得设计组在完成所有的仿真组合之前，就要将设计产品交付出去。而另一个问题是芯片在被设计时，不断发展的芯片加工工艺使得晶体管的特性参数有或多或少的变化，设计工程师需要经常考虑到这一因素。然而这些防范是不足以弥补在硅片上产生的不良的后果的。

即使工艺对芯片核心版图进行了优化，DRAM存储器依然很容易受到加工缺陷的影响。因此，存储器单元、字线驱动器、灵敏放大器和Y译码器正因为这个原因特别容易失效。存储器的设计者通常用冗余电路来替代失效电路，从而提高制造工艺的整体效率。

而对于已经完成的DRAM产品的修改，通常采用程序化的激光熔丝。DRAM的设计包含了冗余的字线驱动、灵敏放大器和Y译码器单元，一旦检测出故障，它们就能被激活。通过激光“吹”熔丝，切断与芯片失效部分的连接，并用冗余部件替代。

激光熔丝的另一个作用是对加工后的芯片进行电路调节。类似于连接选项和金属线选项，激光熔丝也用于调节芯片的运转和性能。

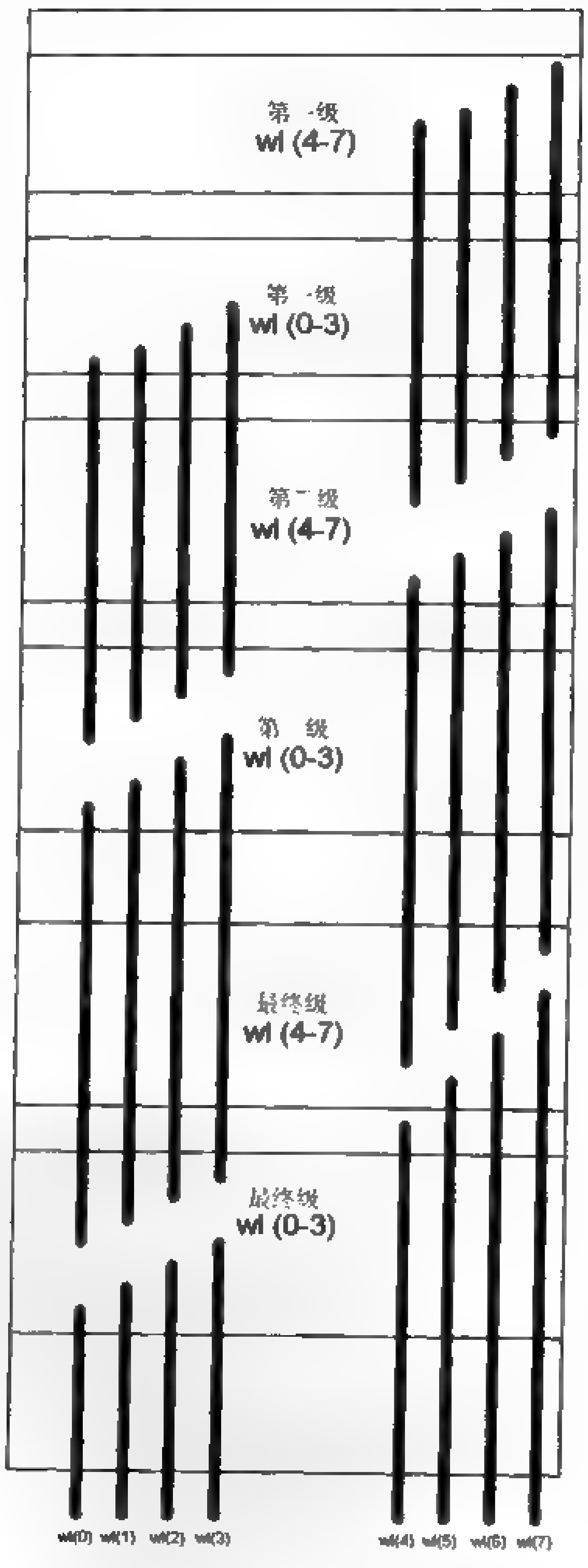


图 5.27 字线驱动器布局结构的示例

熔丝通常用多晶硅和金属线制成，并且必须以能够使激光修复仪器精确地把它们吹断的方式来建制。我们可以想像到，熔丝必须经过特别设计，以避免芯片激光溅射时对内部电路的其他部分造成影响。这些区域需要至少暂时被暴露，所以这段时间很容易被沾污。

PAD单元的设计，需要满足一些物理要求。例如，熔丝必须足够大以利于激光修复器准确地对它们进行调整。熔丝版图的一系列规则如下：

- 熔丝等间距分布的设计易与激光修复器兼容。这些机器到达起点后，便以恒定速度移动；因此，等间距熔丝是最理想的。
- 同样，最小化熔丝的排数减少了把激光移到新起点的操作。熔丝应被设计在活动排组里并且尽可能彼此靠近，这样就能使射不到熔丝的激光头移动的可能性最小。图5.28给出了详细的介绍。
- 熔丝数量需要优化，以使每个芯片修复的时间达到最小化。
- 不明确因素应该清晰地以文件形式列出来，这样测试人员可以准确无误地给机器做规划安排。
- 激光修复器需要特殊的对准图形，确保激光从起点开始自动执行精确对准。通常，这些图形必须密封在嵌入所有熔丝的区域内。

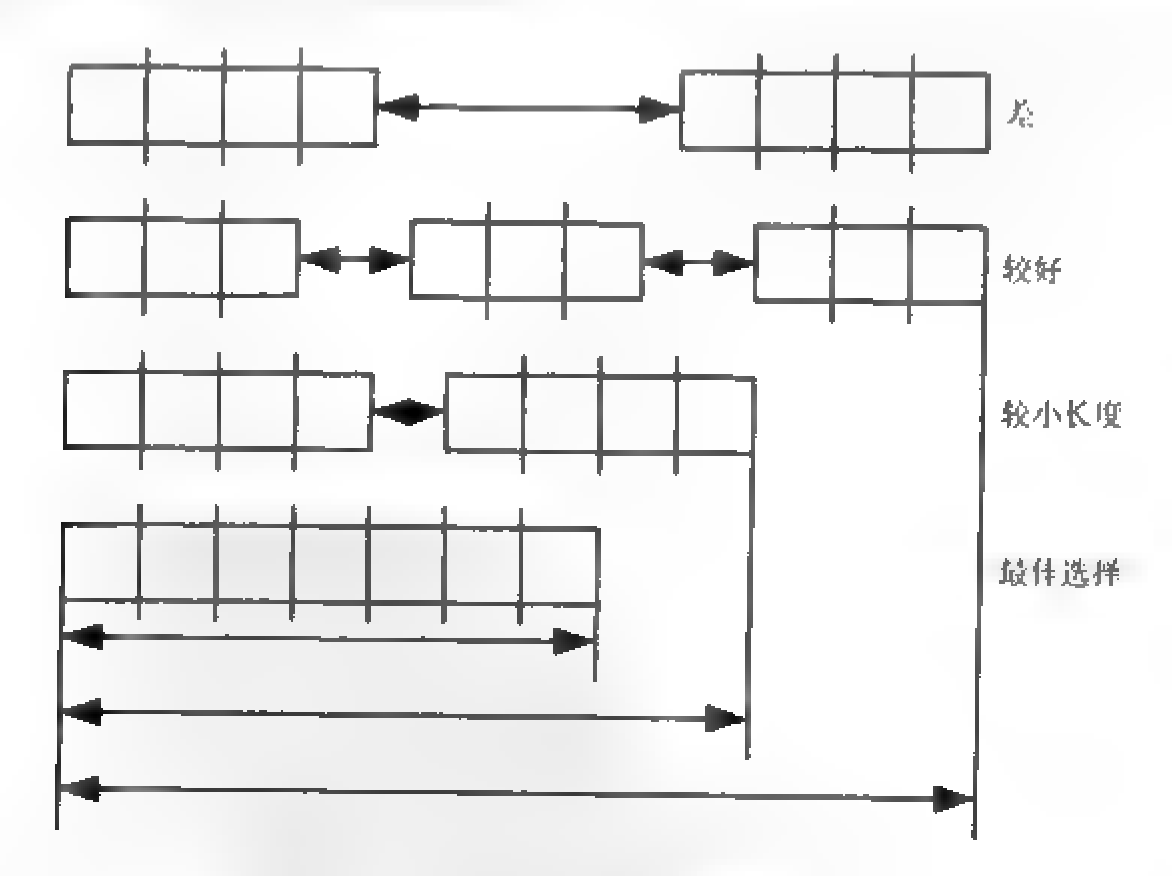


图5.28 熔丝排及其规则

注意从MOSAID设计中提取出的本书附带光盘中的彩色版图，一个是DRAM工艺、另一个是ASIC工艺。图5.29给出了另一个示例。

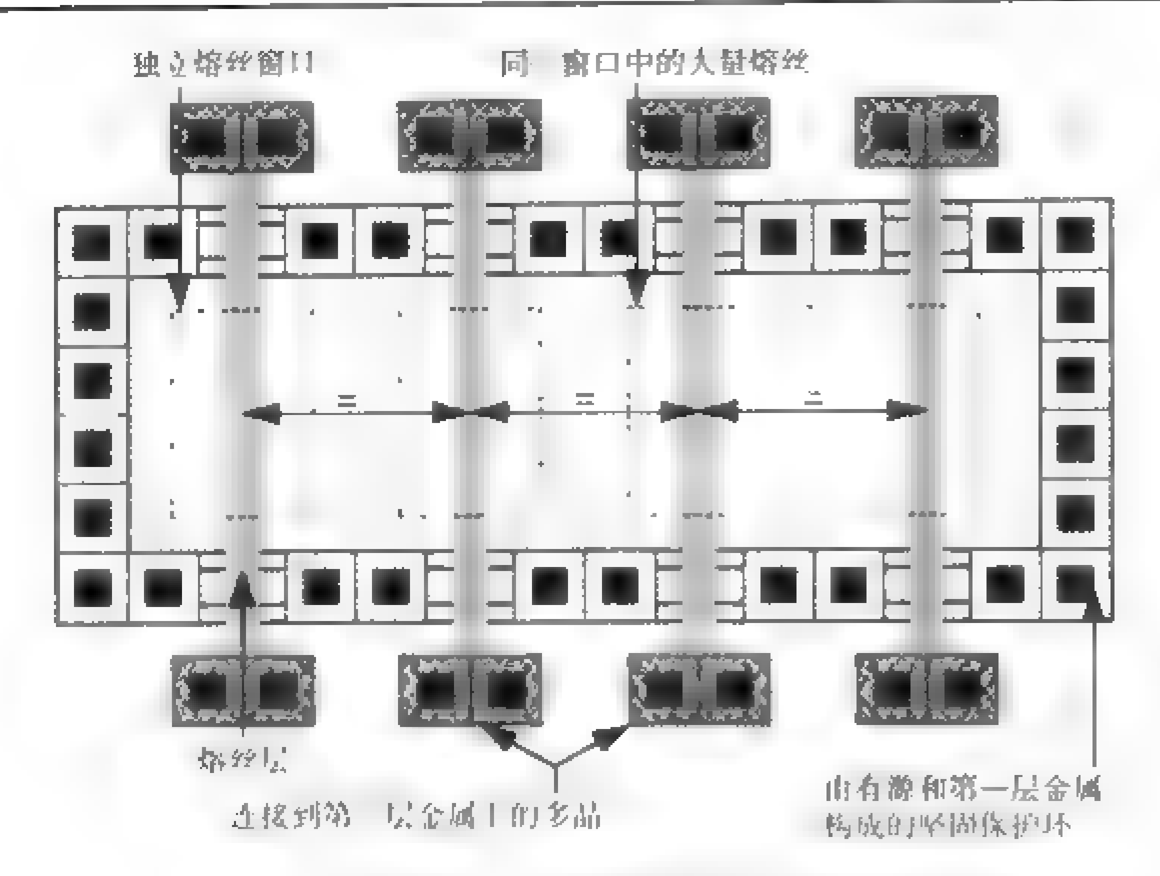


图 5.29 激光熔丝版图示例

5.6 芯片结尾单元

与芯片逻辑功能相关的所有器件在经过布局和验证后，还需要实现一类单元来完成整个芯片并确保其与制造工艺的兼容。

芯片结尾单元的示例包含如下：

- 工艺加工对准图形
- 工艺监控、切片及封装的结构
- 便于肉眼分辨的识别标志：标识、设计首字母、掩模权和拷贝权标志、工艺标志、层标志，见图 5.30。

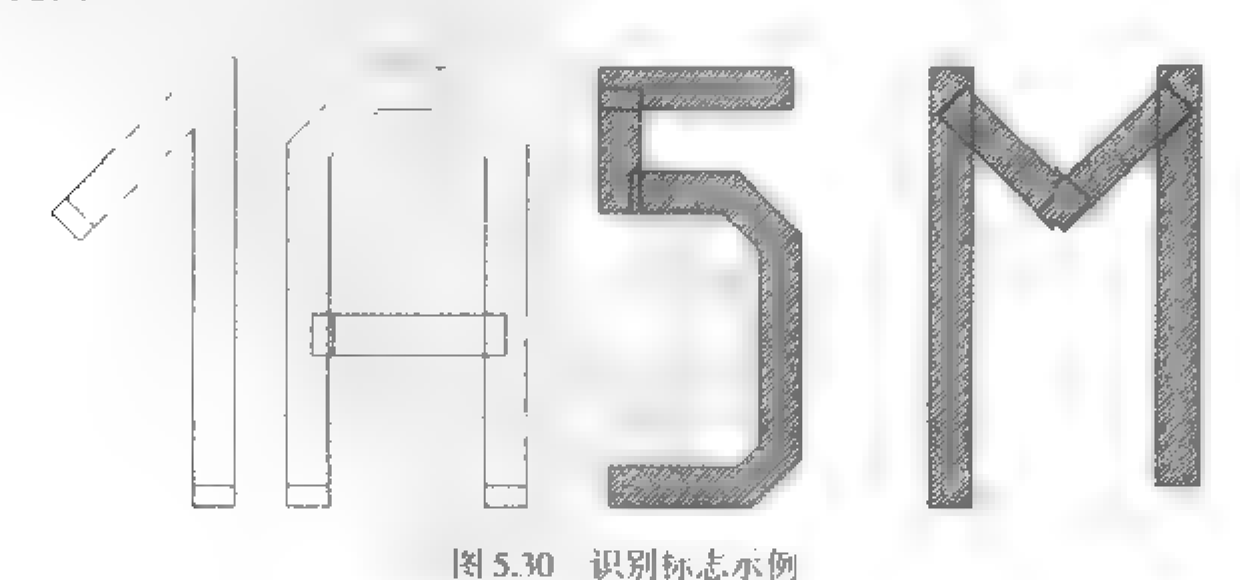


图 5.30 识别标志示例

多数情况下，以上单元由加工厂来完成，并不需要电路设计师参与，但至少版图设计师应该提前获得以上信息，以便为所需器件分配版图空间；否则可能会产生芯片上放置空间不足的问题。

值得注意的是，在所有的器件被分别布置和验证后，整个芯片的最终验证应该包括所有完成的芯片单元，它们包含在掩模制作中。当把识别标志放置在错误的位置时很容易导致电学问题的产生。

5.6.1　对准图形

在版图设计中有很多对准图形，取决于工艺和加工要求。图 5.31 给出了对准图形的三个示例。

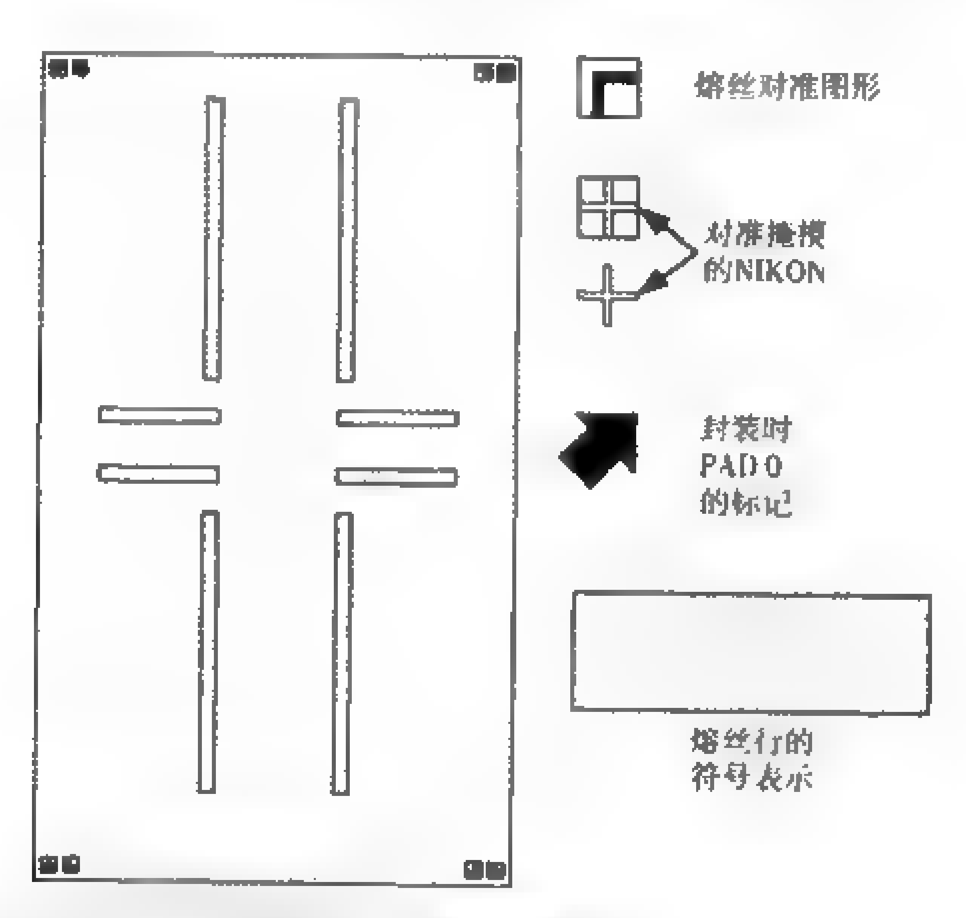


图 5.31　对准图形的示例

激光熔丝对准图形需要至少三个实例。对准图形由在芯片级可见的不同层构成，取决于加工厂商。金属层和通孔层是最常见的选择。

NIKON图形是在晶圆上生成掩模时用来对准刻线的。每一层都放置在这个图形中，因此，这些单元将不能通过版图验证，因为它们会产生设计规则错误和非法元件。NIKON 图形放在芯片的四个角落并尽可能地接近角落。

这里只解释与熔丝和封装相关的几种不同的图形。很多图形是工艺人员进行原型操作和自动机械大规模生产时用来监控制造工艺状况的。

有许多工艺监控器件被用来测量晶体管性能、掩模对准、层电阻和电容，以及层宽度和间距。

工艺监控器件的放置是制版过程的一个功能，因此它们是每层掩模板的一部分。如果一套掩模板仅由一块芯片构成，那么监控器件就应该放在芯片的版图内。如果在一套掩模板上有多个芯片共存，监控器就应该放在芯片之间。

在制造的测试阶段，不合格芯片会以黑色圆点标出，切片时它们会被丢弃。

5.6.2 划片槽和保护环

芯片不是一片一片地生产的，它们是在一个很大的硅薄片即晶圆上制作出来的。一旦晶圆加工完成，它们将会被分割成独立的IC。

独立IC间的狭小沟道是通过尖端划开的方式被机械划开的，这个沟道称为"划片槽"。晶圆由钻石刀片或激光的射烧沿着划片沟道来分割，然后晶圆在机械的重压下沿着划痕分裂开来，从而分裂出独立的IC。图5.32说明了这个概念。

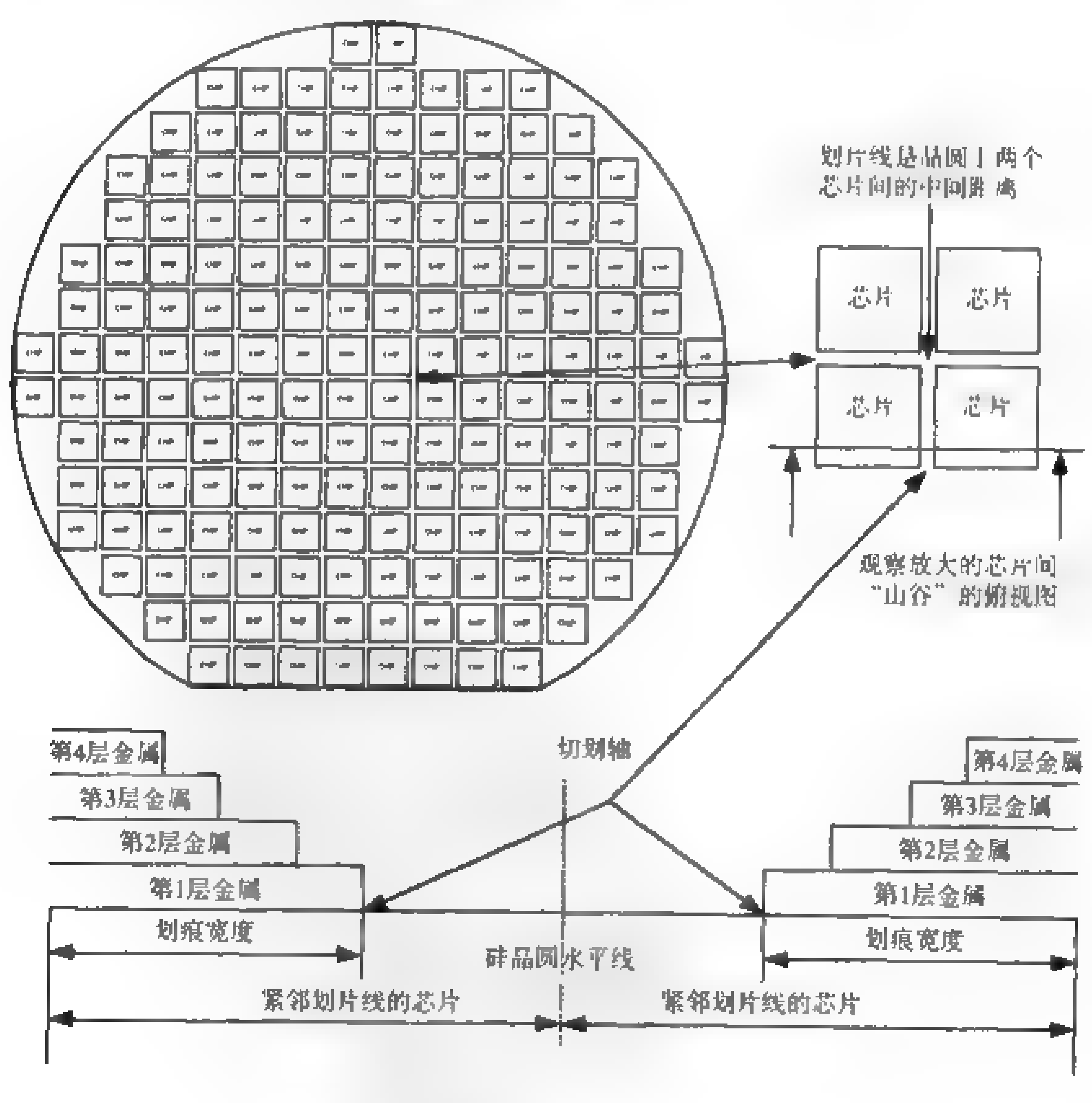


图5.32 划片线

为了保护IC，需要在芯片边缘围绕保护环。

对于一些必须使衬底连接到非标准VSS的源上芯片而言，存在另一种在很多情况下称为“密封环”的保护环。例如，很多包含存储单元的DRAM与标准逻辑单元隔离。在这些情形下NMOS元件放在一个阱中，它与反偏至负压的电压源VBB相连接。存储单元有的内建电荷泵用来生成这种电压及其他电压。如果外部逻辑与VBB相连接，并且整个衬底的外围连到VBB上，芯片周围就形成了一个VBB保护环。这个“保护环”被安置在PAD单元区域之外（但要在划片槽之前）。原因是，放在PAD之外，保护环不会妨碍任何其他信号和/或电路，并且能直接连接到VBB泵上。图5.33是保护环的示例。需要注意的是VBB泵的位置：它应该与密封环尽可能地靠近。

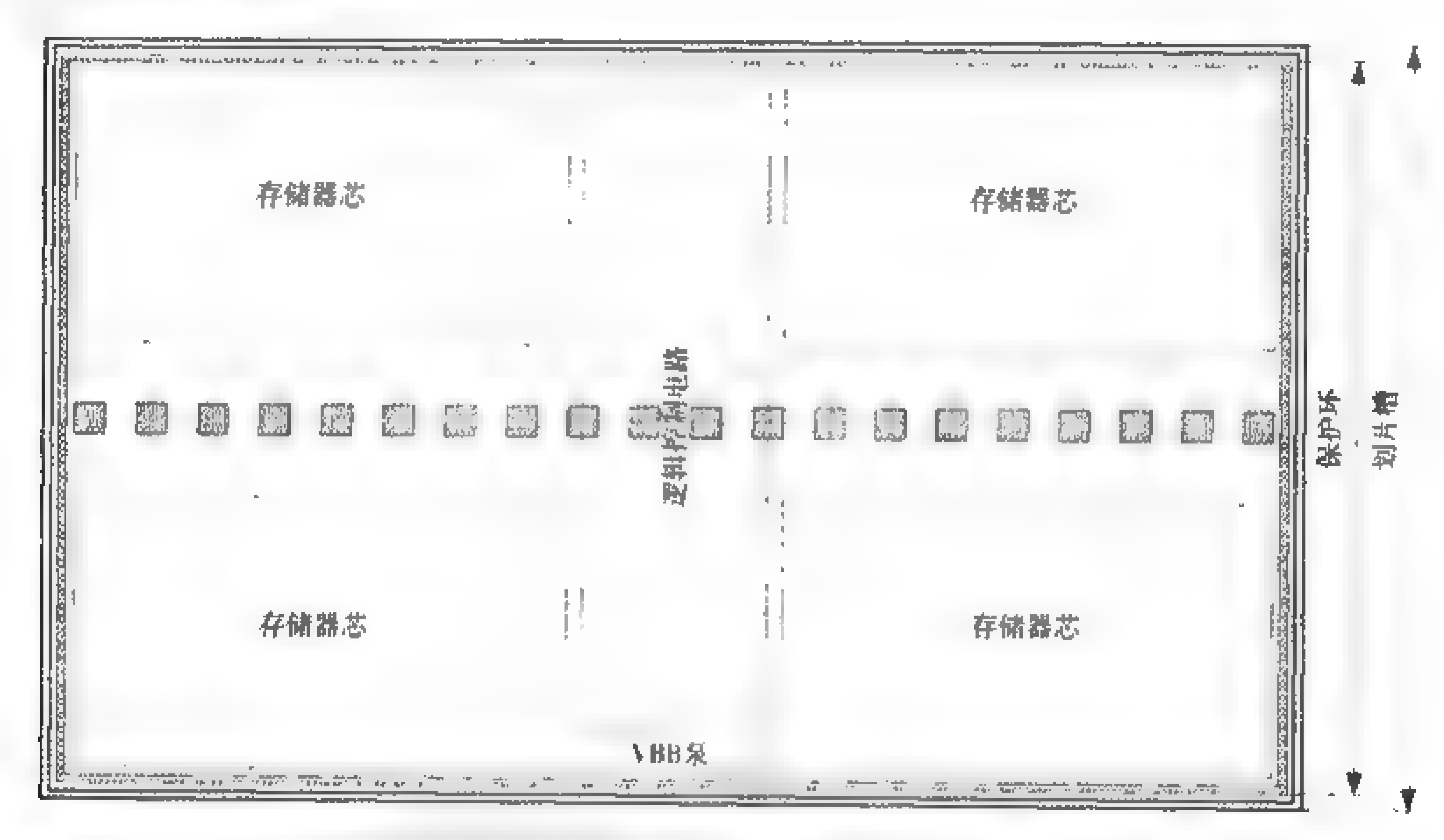


图5.33　密封环

第6章　模块互连的版图设计高级技术

在这一章中，我们将要把我们的版图设计知识应用到不同的互连设计和专用的互连设计中去。互连是设计单元或构建模块之间的连线和连接。

实现不同类型之间的互连有个最佳顺序，下面已经列出这些顺序。那么，这个顺序所包含的原理是什么呢？答案是，我们应该首先实现那些在后续设计过程中最难以进行插入和修改的条目。

从另一个角度看，我们需要首先实现那些互连，应该是占用的面积最大或者是其所影响的芯片区域最多。这两个答案是相互关联的，这是因为这些占用很大面积的互连正是最难修改的部分。

一旦理解了这一点，我们在实现设计中的互连部分时所要遵循的顺序就变得简单明了了：

1. **电源线**：高连通性和大电流要求；几乎连接到所有的晶体管。
2. **时钟**：通常是芯片上最重要且最普遍的动态信号。
3. **数据总线**：需要进行共同布线的一组信号线（即，数据路径）。这些信号合在一起会占用相当多的布线面积并且它们常常是关键路径信号。
4. **特殊信号**：非标准宽度信号或差分对信号、等长信号或一些需要进行屏蔽的信号。实现这些特殊信号的互连就是确保有足够的空间来满足它们的特殊要求。我们将在第7章中讨论这些特殊信号。
5. **一般信号**：余下的所有信号。这些信号就是我们将在6.3节中要讨论的普通的互连信号。

6.1　电源网格

电源线（例如VDD和VSS）是芯片上最为普遍的信号。考虑这样几个特性：电源线实际上会连接每一个门或模块；封装上有多个VDD和VSS引脚；并且它们要传输很大的电流，因此电源线必须要有合理的尺寸。

这些年来，IC设计中对电源管理的需求不断增长。芯片越复杂，其电源网格就越庞大。芯片上的电源电压从过去的5 V降到3.3 V，但是工作频率却增加了。现在芯片内的线网消耗的功耗也在增加。也有许多诸如手机、PDA和手提电脑等低功耗的应用，这些低功耗的应用得益于电源管理技术。在为电源管理提供的计算机辅助设计技术不断发展的同时，低功耗设计也成为了另一个需要研究和掌握的领域。

当为一个设计做规划版图的时候，首先要考虑的事情就是进行电源需求的规划。电源线在规划时要放在模块的周围，并通到模块。在设计完成后，再来增加电源线是很麻烦的，因为通常一个大的、普通的结构需要多个电源线，当设计完成后，就很难再插入电源线了。

记住，增加电源的目的就是提供足够的电源供应以满足整个芯片上全部电路的电迁移的要求和电阻特性。宽的、短的电源线能够满足这两个目标；然而，大的电源线将不可避免地要消耗大的芯片面积。因此，我们在设计的时候，需要折中考虑这个问题。

总之，规划和估计电源线的要求是版图设计过程中必须考虑的部分。

6.1.1 功耗估计

首先要计算每个模块需要的功耗有多大，然后根据功耗需求来估算满足模块需求的最小电源线尺寸。

有很多种方法可以解决这个问题，但其基本出发点就是尽早的解决这个问题，避免在工程进度表的后期有大量的返工。现在的问题就是缺乏很多详细的信息，因为设计工程师不能准确知道在他们的模块中有多少门，也不能准确知道芯片的工作速度可以达到多少。要做到心中有数，这可能是一个不断反复的过程。

一个很好的方法就是基于以前的设计进行功耗估计，借鉴设计的结构，功耗分析的次数和电源的布线。要为新的芯片提供推断数据，我们需要注意以下方面的内容：

- 工艺常数的区别。例如，金属固有的电阻和电容是否有区别，电迁移规则是否有明显的不同。
- 核查所有的通孔和接触孔的工艺参数。
- 可以使用的金属布线层数目，尤其是可以用来走电源线的金属层。层多对走电源线是有利的，层少时，走电源线就会麻烦些。
- 每个模块的速度。不同的模块可能会工作在不同的时钟频率，这将会影响芯片的功耗
- 每个独立模块的规模（规模根据门的数目来判别）
- 基于高速模块引入的噪声，可能需要使用多电源分支结构，用单独的电源线将噪声模块与其他模块隔离开可能是必需的。
- 电源PAD的数目和位置决定了全局布线和从PAD到不同模块之间的通道电阻。

6.1.2 电源布线

假如我们从估算阶段中已经获得了足够多的数据，那么就可以在版图上规划并实现电源布线了。

处理这个问题有很多方法，这里只给出两个基本的方法：

1. **“树根”方法**。在这种情况下，电源线起始的部分尽可能的宽，随着电源供给到不同的模块，电源线的宽度也就越来越窄了，这和树的树干有些相似。

- 根据电迁移因素来决定电源线的宽度，根据每一分支上的电流来确定电源线逐渐减小的比率。
- 当线上所有任意模块电源线电阻无须考虑时，才使用这种方法。
- 在以前，大部分电源线布线工具使用这种方法布电源线。

2.**“阻抗”方法**。这种方法和“树根”方法看起来很像，但是电源线逐渐缩小的比率是根据电源PAD和指定模块之间的计算电阻值来确定的。电阻值的可容忍度是由电源线上可接受的电压降决定的（这些是由欧姆定律计算的）。

- 电源线的宽度是由电源线的电阻率决定的。
- 要准确计算通孔的数量，这可以减少总电阻或让通孔电阻不再是一个限制因素。
- 选择金属电阻率小的金属层作为电源走线。
- 用做电源线的金属层的变化顺序应该是M1 >> M2 >> M1而不是M1 >> M2 >> M3 >> M2 >> M1。
- 在一些特殊或高速的芯片设计中，用多个金属层做电源线，其目的就是减少电源线的阻抗，以确保在芯片上的所有地方都有一致的电压。

图6.1显示了两种基本风格的不同。很多芯片的设计都同时用到这两种方法。

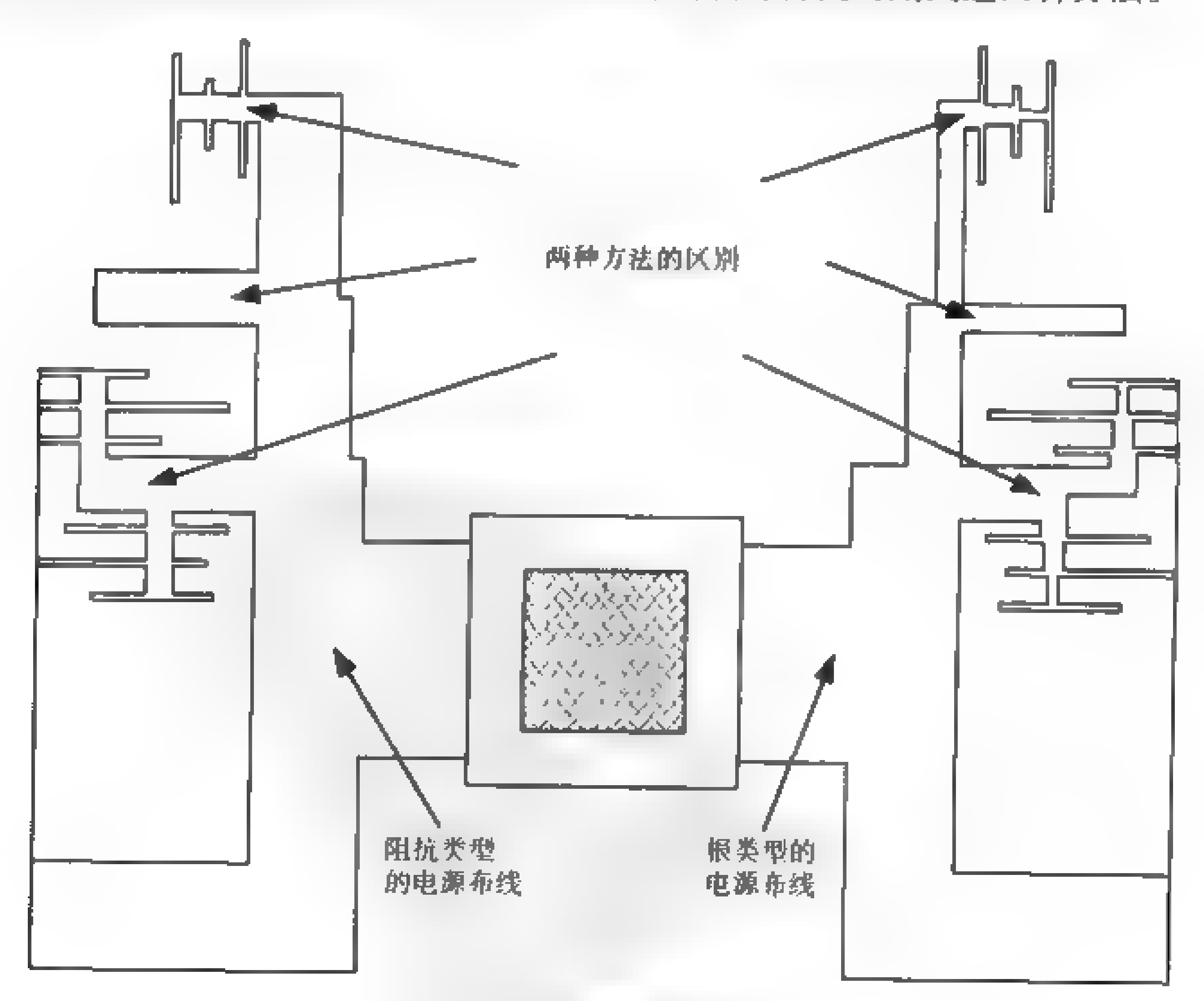


图6.1 电源线布线风格

6.1.3　条和锥化

当我们谈到电源线和芯片版图技术的时候，我们还需要知道“条”概念。我们在第5章讨论字线条单元时，提到过这个概念。

“电源网格”的思想由电源走线和条组成，它们两个共同形成了信号网格。6.1.2节概括了电源信号布线的方法，这一节讲述用适当数量的“条”和“锥化”完成电源网格的设计。在电源布线时，“条”的数目不仅是由电迁移和电阻率决定的，还取决于整个芯片的设计类型和设计的复杂程度。

例如，在标准单元模块设计中，有专门分析一行单元的功耗和电源线电阻率的工具，这些工具自动在单元行之间添加“电源条”来产生固定的电源网格。

某些门阵列结构预先内建了电源网格，垂直的电源条按预先定义的间隔排列。

表6.1显示了在不同环境下等价电源网格的两个例子。条的数目由各种各样的因素决定；这些因素包括功耗、时钟频率、平均扇出、占空比、电迁移和电阻率。

表6.1　电源网格表举例

	行内“strap”的高度	情况1	情况2
单元内M1的宽度	—	2 μm	4 μm
最大“strap”间距	—	120 μm	240 μm
垂直的M2的“strap”宽度	4	6.0 μm	12.0 μm

要全面理解表6.1，应注意以下事项：

- 最大的条间距和单元内的M1宽度之间有一个直接关系，这种关系由电源线的电阻率和到电源网格上的连接定义，假定电源网格的连接是可靠的。
- 这些数字是关于0.25 μm工艺的例子，它显示了不同电源线宽度之间的关系。

图6.2说明了表6.1描述的两种情况之间的差异。

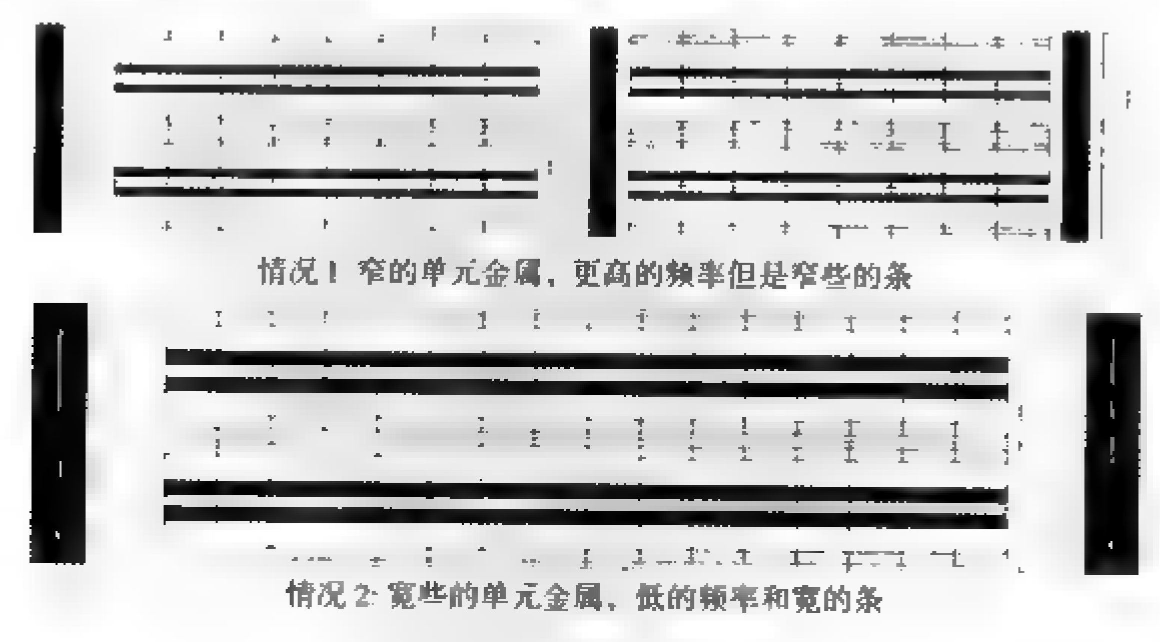


图6.2　等价的电源条例子

从图6.2可以看出，如果对单元有大的布线空隙率要求的话，可以通过牺牲单元高度为代价获得，以增加要求M1电源线的宽度。

6.2 时钟信号

在每个设计中，大多数模块都是在同一个全局时钟信号的同步下工作的。在整个芯片的布线中，全局时钟信号的布线在电源信号之后进行。

由于这个原因，在布好电源线后、布其他信号线前规划时钟信号是很重要和有效的。同样，在整个设计完成后，插入时钟信号是很困难的，这种情况要避免。

从根本上说，实现时钟信号的目的是以最小的延迟把时钟信号分布在一个大的面积上，由于时钟信号有大的电容负载，因此，为了减小延迟，有很多不同的方法可以使用。

全局时钟通常直接源于PAD或是源于内部的时钟产生模块。时钟产生单元的版图设计应考虑如下内容：

- **布局：** 理想情况是把时钟产生单元放在外部时钟PAD和电源PAD附近。时钟产生单元本身是一个大功耗的单元，因此它是电源噪声的一种来源。通过把时钟产生单元与附近的独立电源线或电源PAD相连，来实现噪声和芯片的其他部分之间的隔离。
- **缓冲级的设计：** 正如第5章中讨论的，时钟缓冲单元尺寸可能很大（晶体管的宽度可能是几千个微米）。缓冲链的每一级都要设计成最小尺寸和最小的功耗。为了减少电源连接电阻，使门上的输入电容最小，最重要的是让输出电容最小，在设计晶体管的版图时要使用特殊的技术和方法。

6.2.1 单一时钟信号

实现全局时钟的一种选择是从时钟产生模块那里走单一的时钟互连线。

在这种情况下，类似于6.1.2节中讲述的技术是可以使用的。“树根”布线方法或“阻抗”布线方法在这里都可以使用。

因为时钟信号是动态信号，在这种情况下，电容的影响是比较严重的，需要选择合适的布线层来减小连线的电阻和电容。

遮蔽时钟线有利于把时钟信号和其他的信号隔离，也能减少时钟信号的耦合电容。

6.2.2 时钟树

另一种常用的时钟实现方案称为时钟树。在专用集成电路设计类型中，它是很常见的，因为时钟树的自动生成很容易集成进ASIC设计流程之中。

时钟树是插入在时钟信号路径上的缓冲器网络，它通过插入缓冲器使时钟源和所有终端之间的延迟减小。这并不是优化电学信号通路，而是分开通路插入缓冲器来使延迟最小。这样产

生的网络和树有些相似，中心时钟信号的分支通过使用缓冲器遍布整个芯片，在时钟信号到达所有叶单元处结束。

通常，实现时钟树有下列几个步骤：

1. 首先要完成逻辑单元的放置。这确保了芯片核心逻辑的时序性能得到满足。
2. 考虑逻辑单元的位置，插入时钟树。在合理的位置插入缓冲单元来使时钟延迟最小和走线最短。
3. 完成所有信号的布线，优化布线以满足时序要求。

在专用集成电路流程中，可用的自动化工具有很多种；然而，同样的步骤适用于不同的设计。原则上，设计工程师在设计一个时钟树时，要考虑下面的情况：

- 首先要定义/了解时钟树的范围，包括诸如总负载、布线面积、时钟信号线的长度，可用来走时钟信号线的布线层和布线约束条件。
- 定义时钟树必须满足的约束，包括最小和最大的插入延迟以及最大相位偏差。
- 定义时钟树拓扑布局产生的方法，包括时钟树有几级缓冲，缓冲器-反相器类型和每级扇出的限制。设计工程师可以手动产生时钟树拓扑布局，也可通过时钟树产生工具自动实现。

图 6.3 是关于时钟树的两个例子。

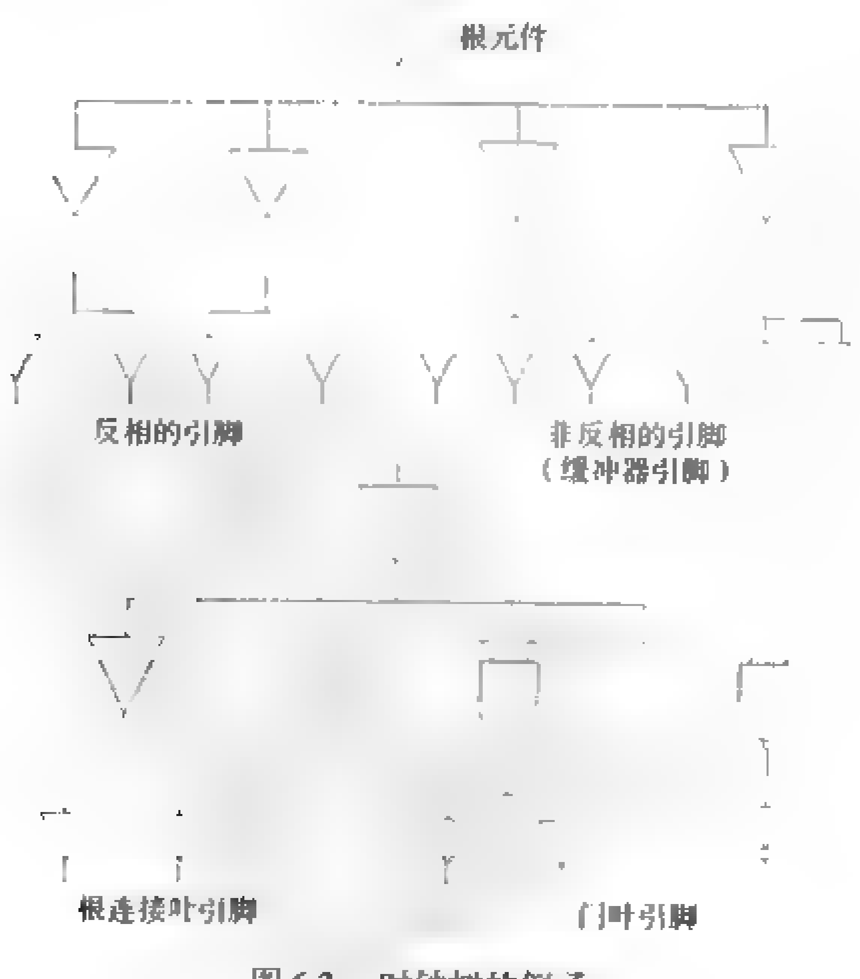


图 6.3　时钟树的例子

6.3 互连布线

当解决了电源布线和时钟树问题以后，下一步就该解决一般布线问题了。特殊信号的布线问题将在第7章讨论。首先让我们回顾一下为信号布线的正确顺序：

1. 电源线。
2. 时钟信号。
3. 总线。
4. 特殊信号线——将在第7章讨论。
5. 一般信号线——本节的主题。

6.3.1 布线规划

布线规划的目的是判断整个布线的复杂程度，确定芯片上用于布线的区域（这些区域也叫布线通道），并找到在完成全部布线的过程中潜在的瓶颈或问题。另外，也要预先估计到布线对最终芯片面积的影响。例如，会估算用于布线通道的面积并包括在版图规划之中。

下面列出了一个可以实现的可用的布线规划步骤：

1. **信号估计**。如果没有最终的电路图或者网表，是不可能确定用于模块之间互连的信号线数目的。不管布线规划是多么完善，它仅仅是对将来的一个预先判断而已，而且它还可能是错的。但是需要说明的是，不管规划是多么粗糙，有总是比没有好得多。

没有最终的电路设计，我们仍要估计信号线的总数。信号线总数对布线规划来说是一种有效指导，假如我们对信号线进行分组或者将信号线与不同的模块联系起来，我们就会对拥塞的区域有个大致的把握。

对信号的估计可以基于如下几个方面：

- 向有经验的、以前曾经做过类似芯片的版图设计师请教，让他基于自己的经验给出一个对面积的估计。他们在一些方面的知识是非常有价值的，比如拥塞的区域，基于不同模块的特性会有多少信号线，或者过去遇到的值得注意的地方。
- 尺寸估算和在设计中为每个主要模块预备引脚列表。
- 信号源和目标点——也就是信号从哪里来到哪里去。芯片架构工程师使用模块化方式设计的仅仅用于面积估算的简单电路图（不是100%正确）对于估算面积也是非常有帮助的。这个电路图为所有的主要模块预留了位置，反映了对每个模块尺寸的估计和它们之间的面积比。图6.4就是这样一个版图规划的例子
- 主要的总线和特殊信号线的列表。
- Pad列表及其在芯片周围的位置。

使用以上所有这些信息，我们能估计：主要布线通道的位置，根据每个通道的信号线的数目计算布线通道的尺寸。

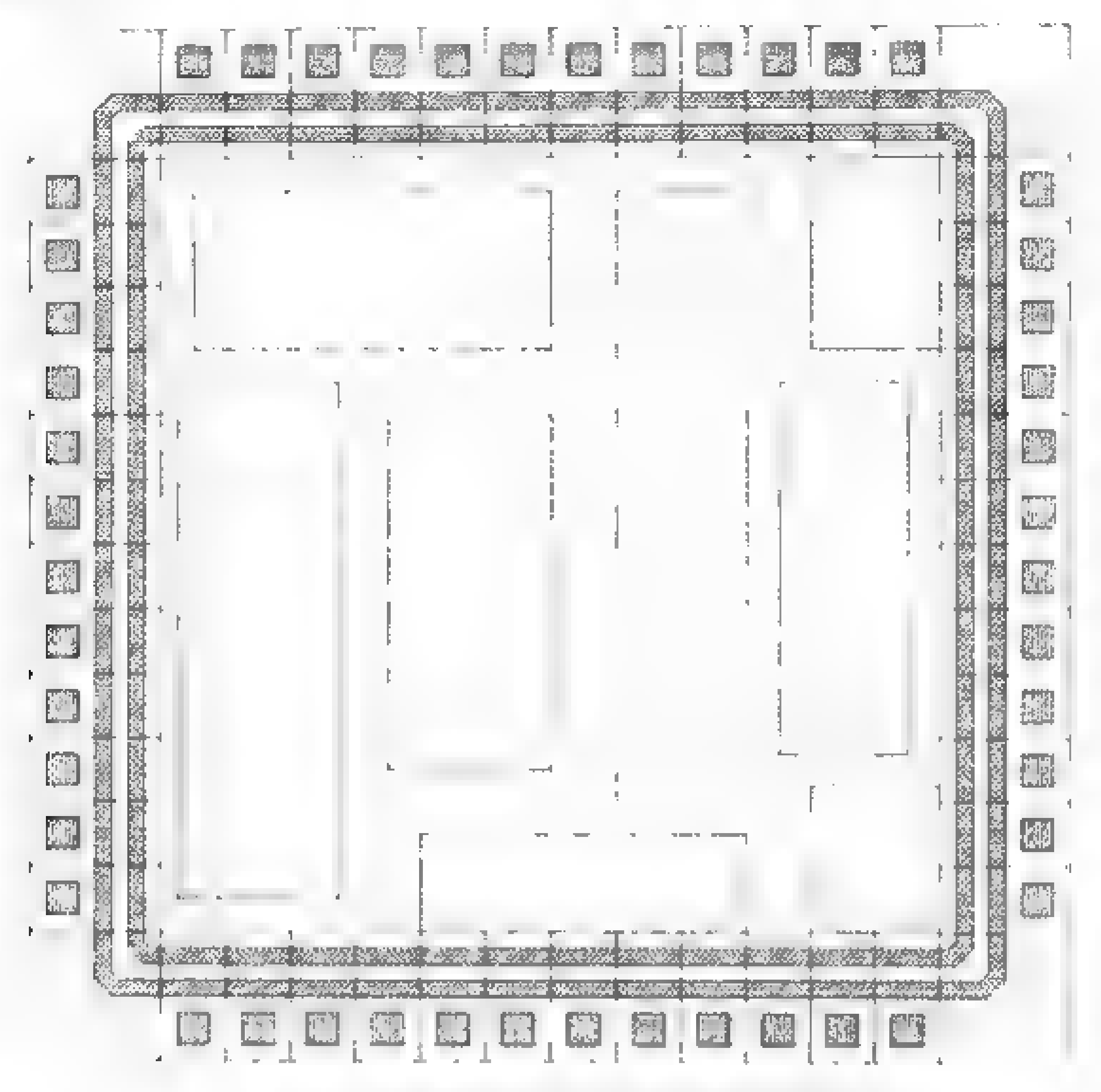


图6.4　表示信号模块和布线通道的版图规划举例

假如模块的上面允许走线，那么我们在定义布线通道的尺寸和位置的时候需要将其考虑进去。在这种情况下，做好层次信号规划并将每个模块的可布通性考虑进去是非常有用的。

2. **估计布线方向。**对于每一个分层来说，要一个通道一个通道地决定布线方向。在版图规划阶段选择最优的布线方向是一个好方法。

在每一个通道中分层的不同使用情况也应该加以考虑。

这些问题将在后面的章节中做进一步阐述。

3. **偶然性规划。**最后，为处理设计中产生的重大改变而带来的特性和开销应该纳入规划。例如，备用的线和备用的空间应该纳入规划。为处理重大改变而带来的开销总量基于下面几个因素决定：

- 设计的新颖性。对于新设计，在预估模块的尺寸、信号线的数量和PAD的位置方面，往往会有更大的不确定性。

- 工艺的稳定性。工艺改变以后，布线层的设计规则将会改变，有时变大，有时变小，这将会对布线规划产生显著影响。
- 对于一般复杂程度和新颖性的设计来说，10%的偶然性规划是个可行的规则。

4. **监视和更新。**当电路设计成熟以后，由于模块和全芯片的更多细节是可用的，所以布线规划应该被不断更新。

这是一个处理在实际布线规划中遇到的所有需要关心问题的过程。关于芯片的尺寸、拥塞和可布线性的问题将会暴露出来并在早期阶段处理。一旦电路设计确定下来以后，布线规划将会使设计的最终实现变得更为简单并减少出错的可能性。

6.3.2 通道顺序和布线方向

既然我们已经在布线规划中确定了布线通道，那么就可以更详细地检查单个通道。每个通道被确定为包含一定数目的信号。

我们到底关心什么？本质上说，规划布线通道的目的是为了确定总线信号的优先次序，用以优化下面的几个指标：

- **电路性能方面的需求：**包括关键路径信号，信号线的电阻和电容。
- **通道的面积：**布线通道可以被优化以防止短线连接和不必要的层次改变，尤其是一个通道的信号进入另外一个通道。

实际上，在大部分情况下，假如布线通道的面积被优化了，电路的性能会同时得到优化。手工实现和优化布线通道的简单步骤如下：

1. 为通道中的每个信号添加一个无命名的路径
2. 假如通道中有已知的重要关键信号，首先给它们做上标记，并决定它们在通道中的位置
3. 标记并放置横跨通道全长的信号线。
4. 标记并放置在通道中开始或者结束的信号线
5. 假如知道有信号线简单地绕过一排逻辑到毗邻的通道，可以考虑添加直通单元来实现。关于这个问题，将在6.3.3节详细讨论。
6. 使用剩余的空间标记并放置局部信号。局部信号是在同一通道内开始并结束的信号。
7. 如图6.5所示，注意单元的布局可能会提高局部互连线可用性。参考图6.6的例子可以看到布局对通道尺寸的影响。
8. 在通道中留下一些空间以满足新的或可能的需要。
9. 当信号绕过边角从一个通道到下一个通道时，如果有必要对通孔数目或分层改变进行优化或使其最小化，那么应对信号线进行重新排序。图6.7是一个理想的通道顺序，其减少了需要改变通道的信号线所带来的开销。并不是所有的布线工具都具有这一功能！
10. 打印一个完整的设计，以确认为优化设计所做出的改变。

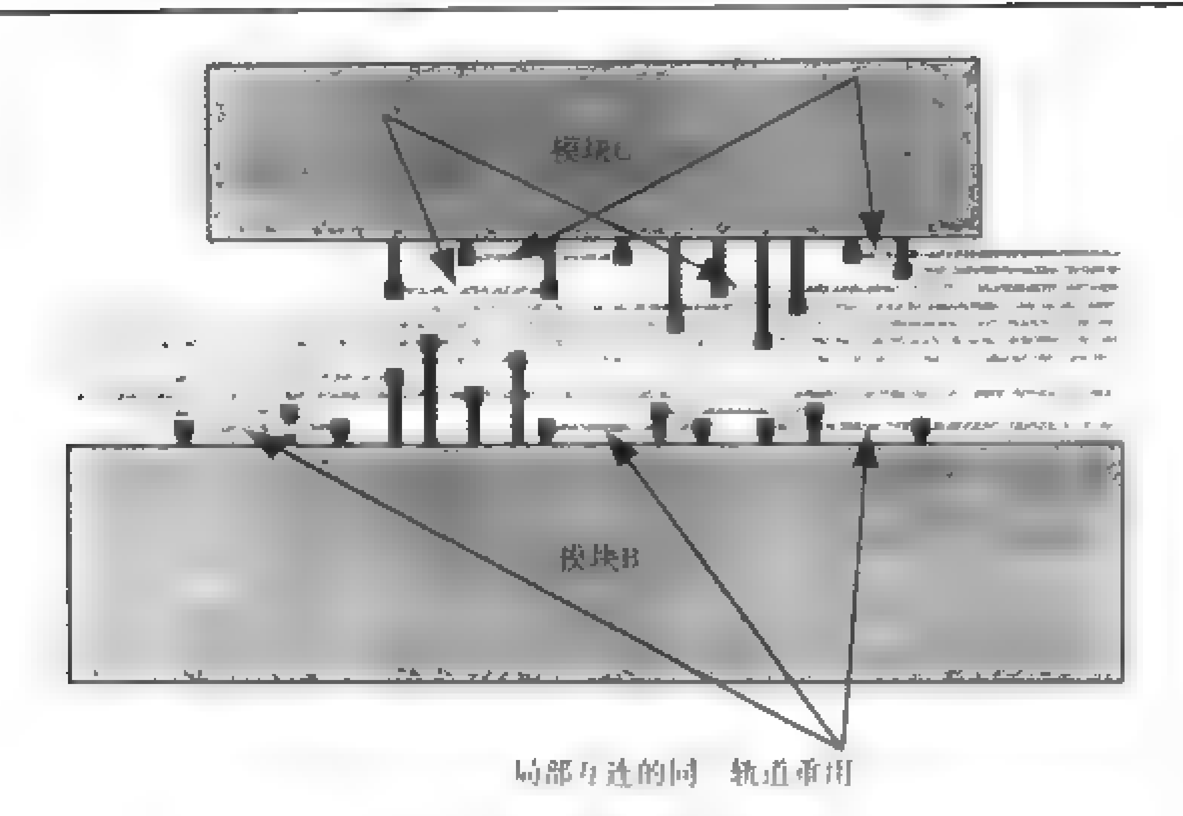
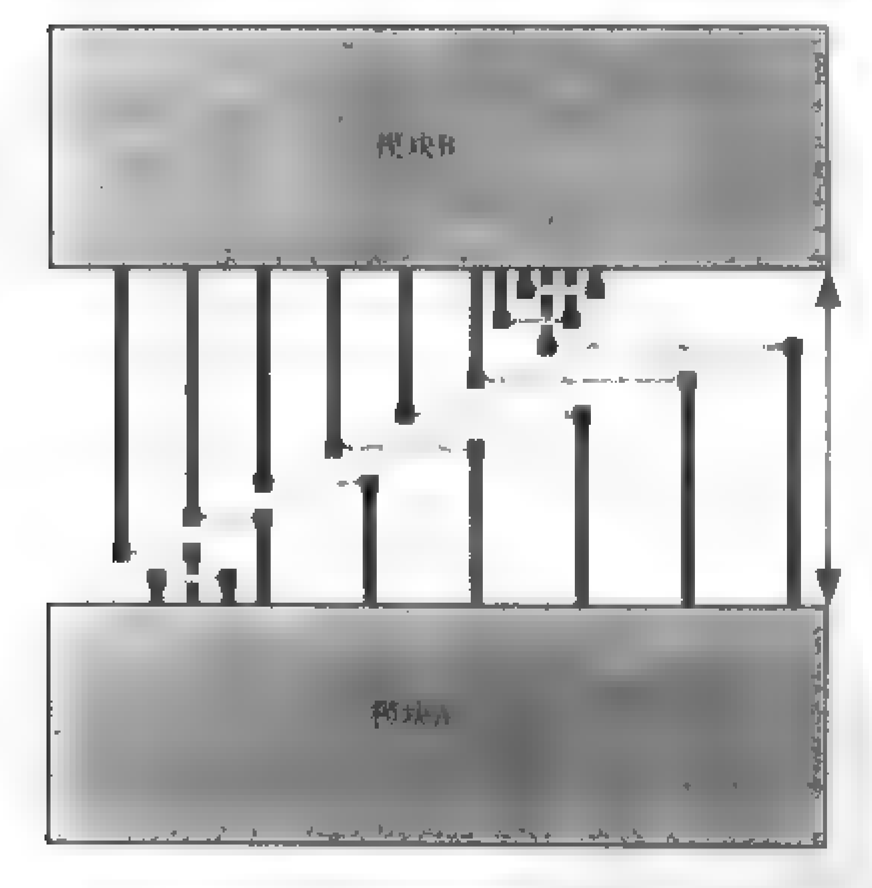


图6.5　局部布线共享



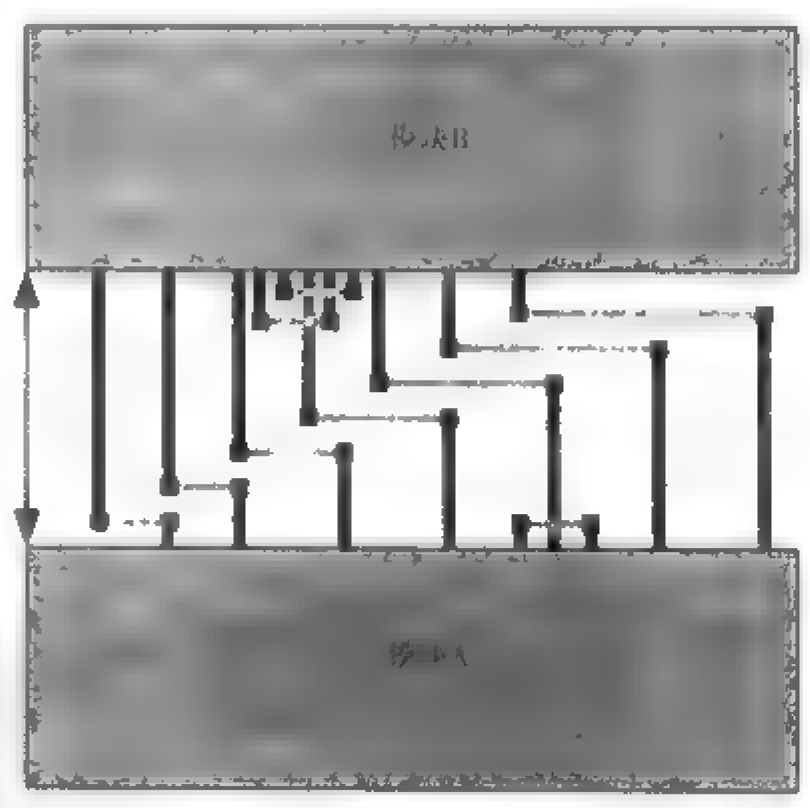


图6.6　改变单元位置以优化布线通道

正如我们已经提到的，对于每个布线通道，其中的布线层应该被单独决定。为整个芯片定义布线层的方向和教条地使用某种方向都是不必要的。正如我们所看到的，为每一个布线通道明智地选择布线层将会使布线通道的面积得到优化，这依赖于对信号线的布线需求。

对于电源线、特殊信号线和宽总线来说，每一层的布线方向应保持一致，这类信号是全局性的，有利于层次分配的标准化。

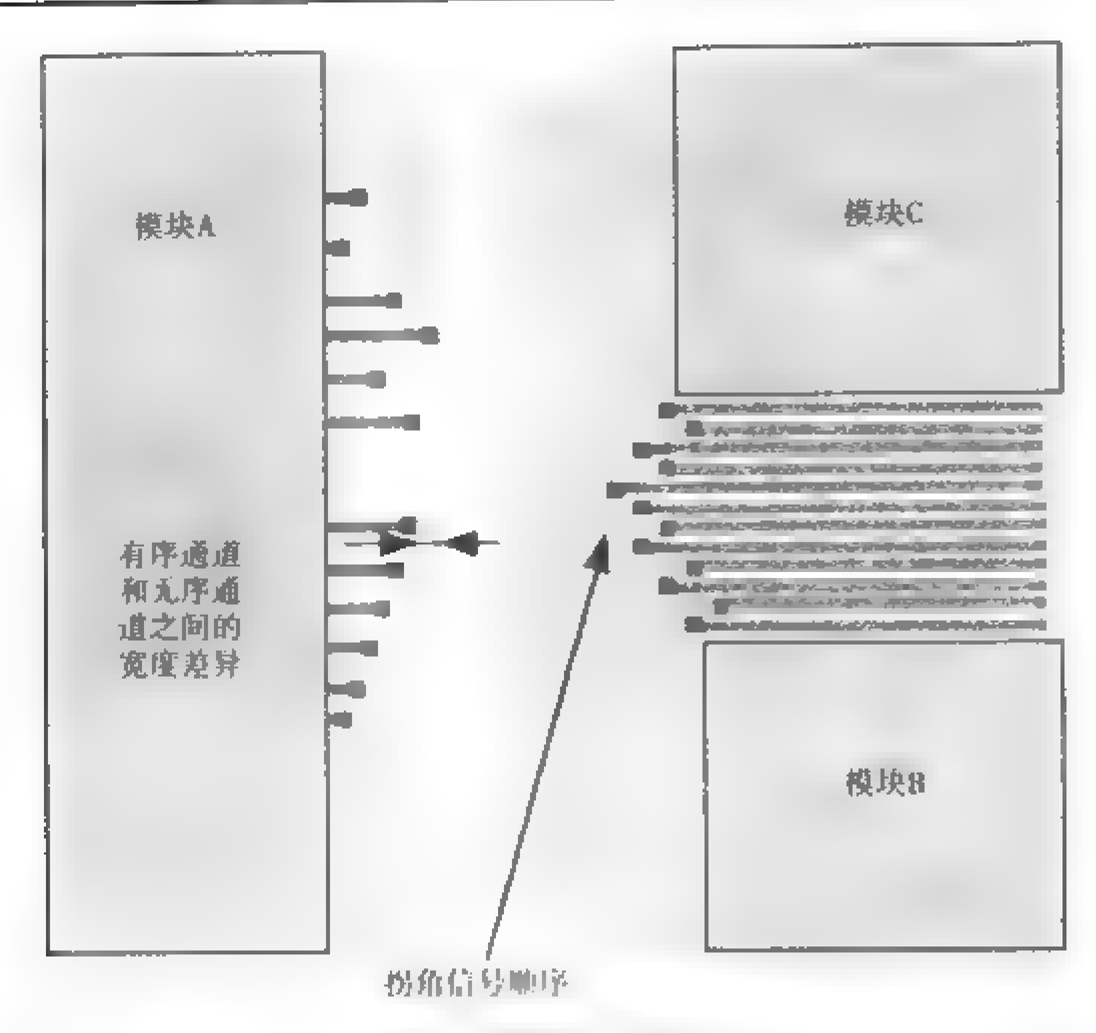


图6.7 通道顺序举例

在一个通道内部，决定最优布线层的关键因素如下：

- 如图6.7所示，需要为转弯的信号线合理分配顺序
- 单元的引脚使用的分层将决定通道中与单元连接的信号线使用的分层。注意，单元的分层选择也是重要和需要加以考虑的，因为假如一个特定的通道布线层使得版图的面积更小或者效率更高，那么单元设计应该利用这个因素
- 一般来说，信号线布线要与单元行平行，并且其布线方向应与单元内部同一分层上的信号线相同。
- 穿越通道的局部信号有多个布线层可供选择。图6.8给出了关于这种情况的三个例子。我们可以看到，在模块B中的信号和模块C中的信号顺序相同，仅仅是位置不同而已。这是一个布线层用于所有方向的典型例子。这样的方法使所述的总线获得了最佳的版图面积和总线对称性

对图6.8，需要注意以下几个方面：

- 单一层布线只有在穿过通道的信号的顺序维持不变时才是可能的
- 通道2和3不必像通道1那样打孔。

- 在通道 1 中，走线在两个布线方向上都受到了阻塞；在通道 2 和通道 3 中，垂直走线不受约束和限制。
- 通道 3 仅仅使用一层来实现，但实际上，在用阴影显示的布线中，通道 3 和通道 1 的尺寸是一样的。45° 布线并没有节约面积，然而却消耗了更多的布线时间。
- 通道 2 是三个之中最佳的版图实现方案：版图对称，没有垂直布线限制，布线通道的面积小于其余二者。

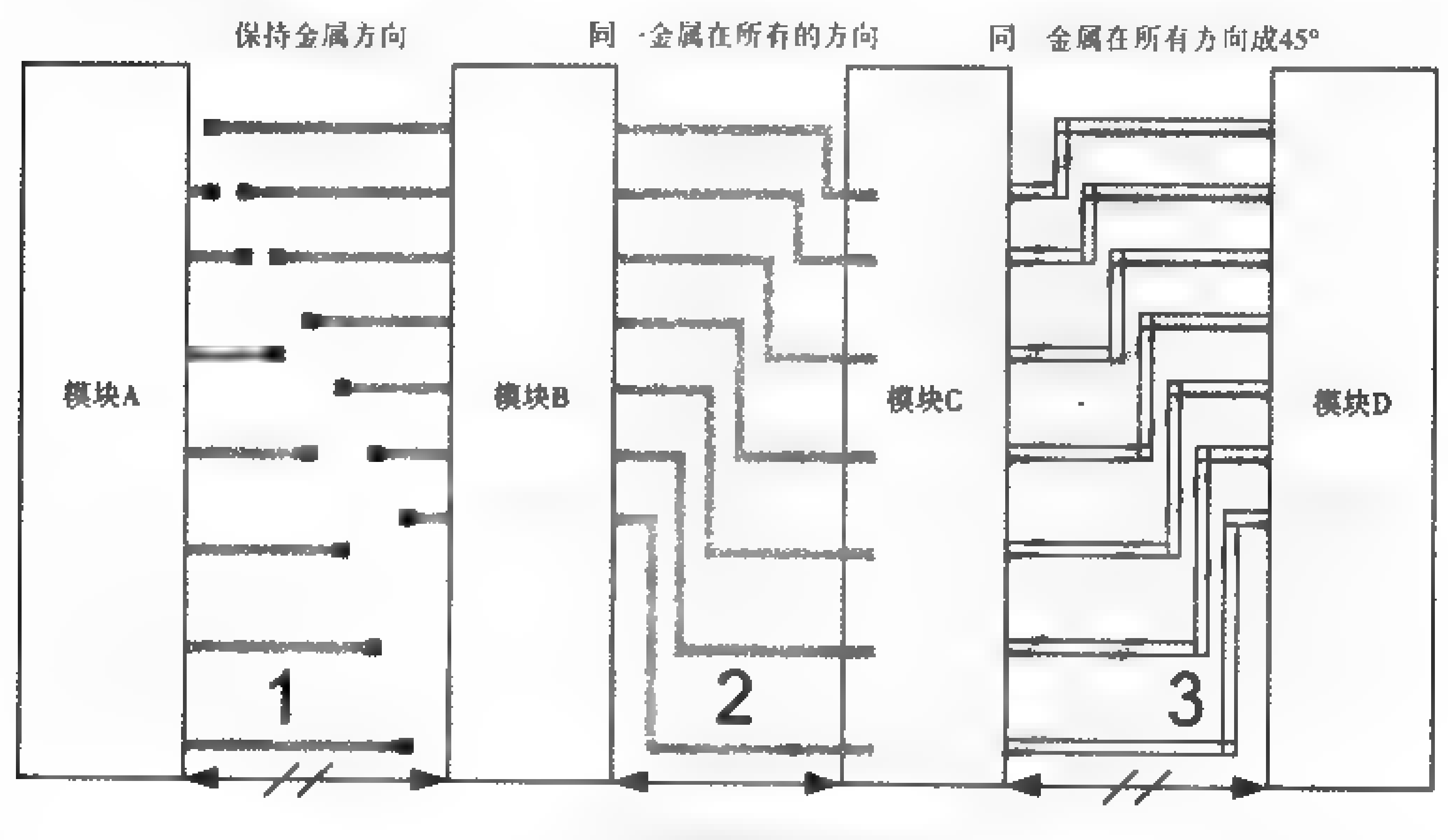


图 6.8　跨越通道布线举例

6.3.3　使用直通

顾名思义，直通是一种仅穿越结构而无任何电气连接的布线通道。在第 5 章中曾对将其作为一种信号穿过标准单元行的方法进行了介绍。

现在，让我们来看看在设计中包含一些模块和通道时直通的使用情况。对版图设计而言，版图规划是至关重要的；并且由于版图规划允许我们对信号布线进行分析和优化，其也是一种非常有价值的设计描述方式。优化工作甚至可以在我们完成模块内部版图之前就得以实现，而布线分析在此时最具价值。布线要求能够得到预见，并且可以在模块设计中实现。因此，版图规划所带来的好处决不能低估。

考虑图 6.9 的情况，在图中，从模块 A 到模块 B 的一些信号线环绕在模块 C 的周围。解决这一问题的方法有很多，我们首先假定模块 A 和 B 是已经完成的设计并且不能改变，而模块 C 还在开发中并可以更改。

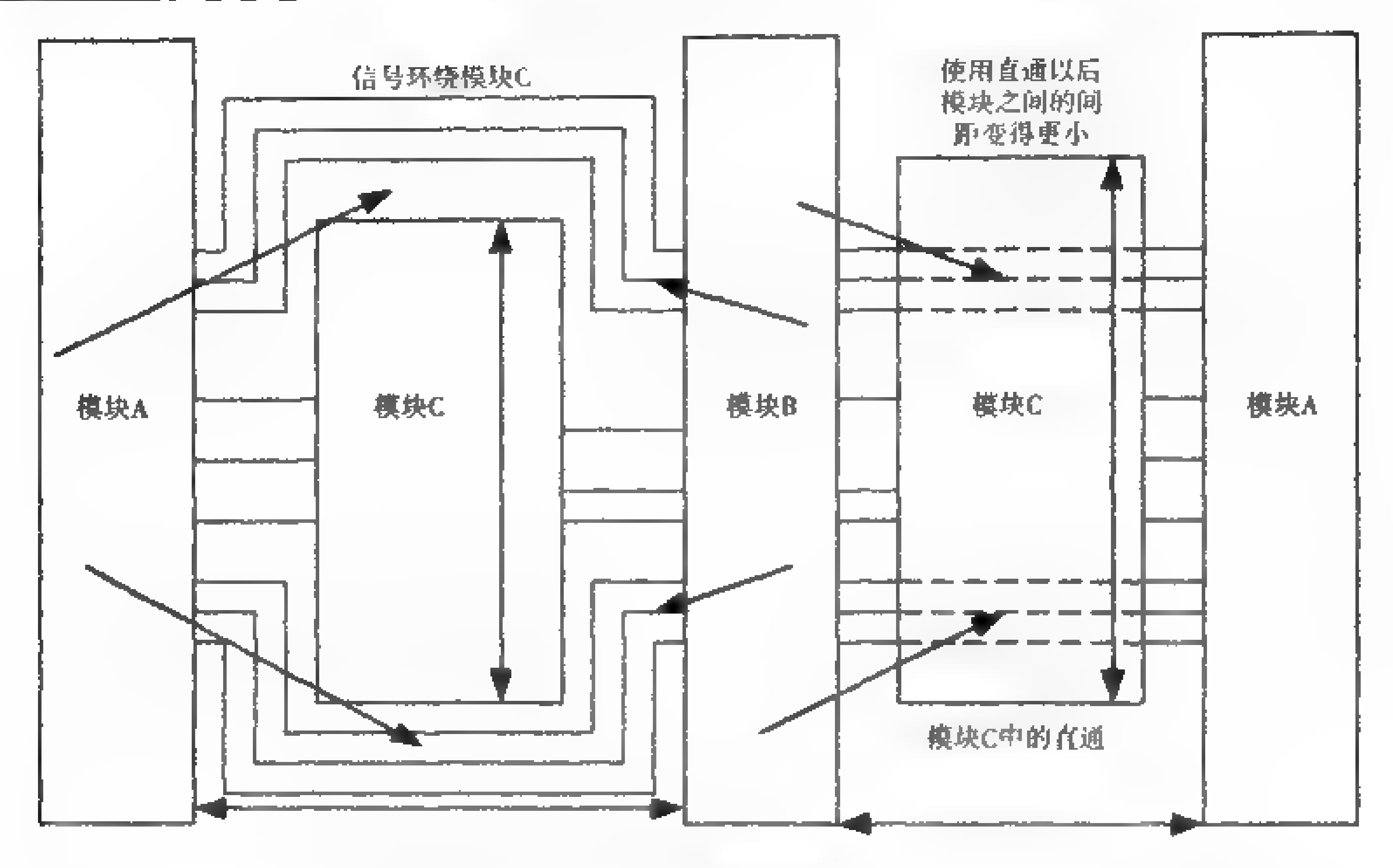


图6.9　直通举例

图6.9中的右侧给出了在模块C中增加直通的效果。

- 在这个例子中，模块C被拉长以容纳额外的信号线。这表明在此方向和期望层上都没有足够多的布线资源来容纳这些信号线。
- 这些从模块A到模块B的信号线变短了。
- 模块A和B的距离变近了，这是因为其与模块C间的通道显著减小了。
- 模块C变高了，因为在高这个维度上增加了信号线，这种情况发生在有足够的布线资源处理额外信号的时候。
- 模块A、B和C的端口必须对齐。
- 模块C的验证需要包括额外信号的结果，一种很好的做法是给这些信号打上标号以确保这些信号被作为直通信号看待，这样就不会被识别为另一种方式。

在很多情况下，版图需要做很多的改变和增减。当版图完成后，有很多版图设计方法可以使用直通的思路进行更改。这些技术也可用于正在被实现而没有版图规划信息的底层模块。

在对额外的布线进行预估或者将空余的布线资源纳入模块设计时，需要注意以下几个方面的问题：

- 这个方法可能会影响模块的空隙率，并增加单元面积的开销。与此同时，还不知道布线空间是否会被使用上。

- 适当地标识出没有用到的布线资源，以免忘记。
- 标识出不可用于布线的区域以防止发生错误。
- 确保模块的性能没有受到所增加的多边形的寄生电容的影响 类似地，假如布线资源没有用到，应对模块的性能加以验证。
- 处理这类问题的最简单的办法是将布线资源作为单元的 -部分看待，并对带有额外信号线的模块的验证开销进行处理 在这种情况下，模块的特性是最可预期的

另 -种方案是将模块设计为几个部分，它们之间很容易分开，以容纳额外的信号 图6.10给出了这种实现方法。

注意，通过使用这种方法，我们能够增加电路和布线资源

需要强调的是，假如使用了版图规划，最后的两种方法将很少被用到！这 -技术对于小版图、大单元或者模块是很有用的

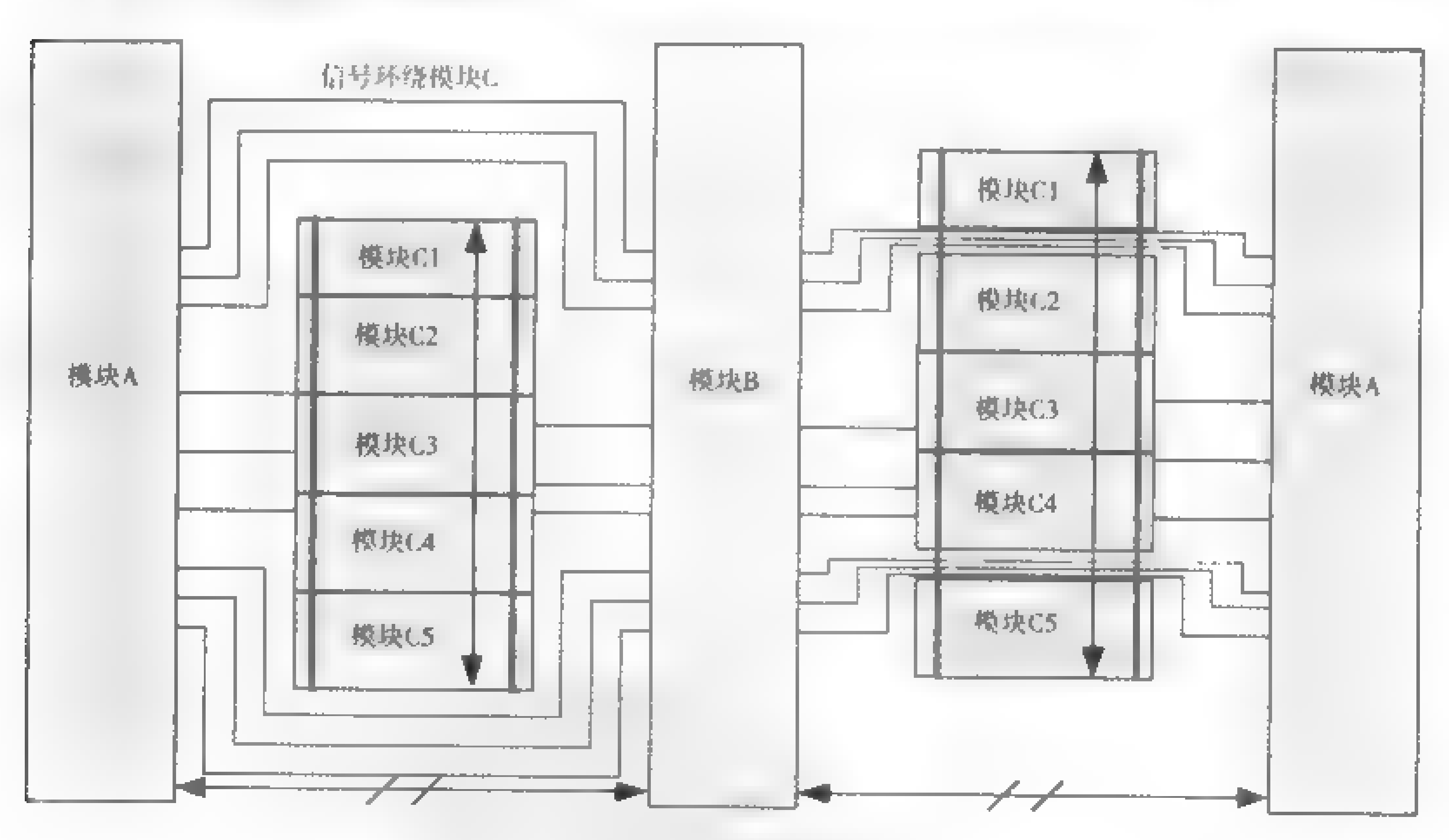


图6.10　为直通劈开模块

第7章　考虑电气特性的版图设计技术

在这一章中，我们将讨论一些在电路设计中对电气性能有直接影响的问题，这些电路设计将要在版图上实现。我们的讨论包含以下方面：

- 电阻和电阻器
- 电容和电容器
- 对称和平衡
- 更多高级版图设计技巧

7.1　电阻

在IC设计中，电阻计算的惯用方法是，以每平方电阻为单位来描述每个导体层的阻值。当导体的长等于宽时，即定义为一个“平方”。计算一个导体电阻的公式为

$$R = \rho \times l/w$$

这里，ρ表示导体层的电阻率，单位是Ω/□（读做“每平方欧姆”），l表示长，w表示导体的宽。

从导体电阻公式可以看出，显然有两种使一个多边形电阻最小的方法（ρ是制造工艺的一个特性，不能由版图设计工程师控制）：

1. 减小多边形的长度。
2. 增加多边形的宽度。

复习串联和并联电阻的等效电阻的计算方法，对理解如下问题是十分重要的，即在各种版图设计风格中，什么时候好的设计习惯可以减少电阻。

图7.1列出了以不同方法连接的电阻的例子，以及在A和B两节点间的总电阻的计算。重要的是要注意到，串联电阻是累积的，而并联电阻减小了有效电阻。

7.1.1　在晶体管设计中使电阻最小化

在第6章中，我们讨论了布线方面的电阻。现在，我们考虑在晶体管级版图设计中的电阻影响。

在CMOS工艺中，一个晶体管是由源（有源）区、栅（多晶硅）区和漏（有源）区组成，为了使晶体管工作，我们需要将信号线连接到全部三个端上。因此，考虑源和漏上的接触是很重要的。

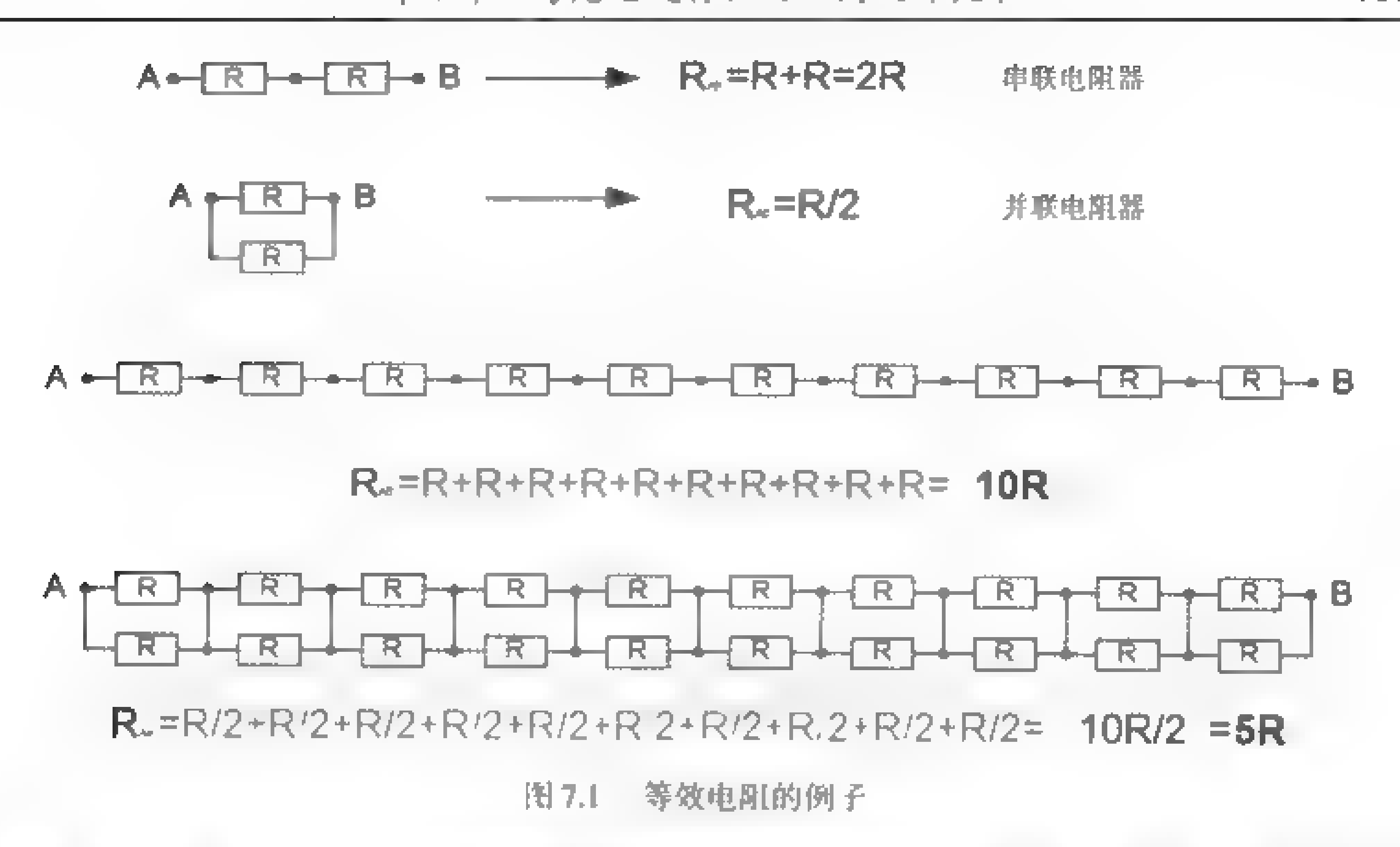


图7.1　等效电阻的例子

图7.2给出了一种含有不同类型电阻的复杂的晶体管电阻模型，其描绘了横跨整个晶体管的许多不同的电流通道。在多晶版图中给出的每个电流弧，在电阻模型中都有描绘。这个图例给出了每种类型电阻的近似值，并说明了在电路中每个电阻的相对阻值。这些阻值是0.25 μm工艺下的典型值。

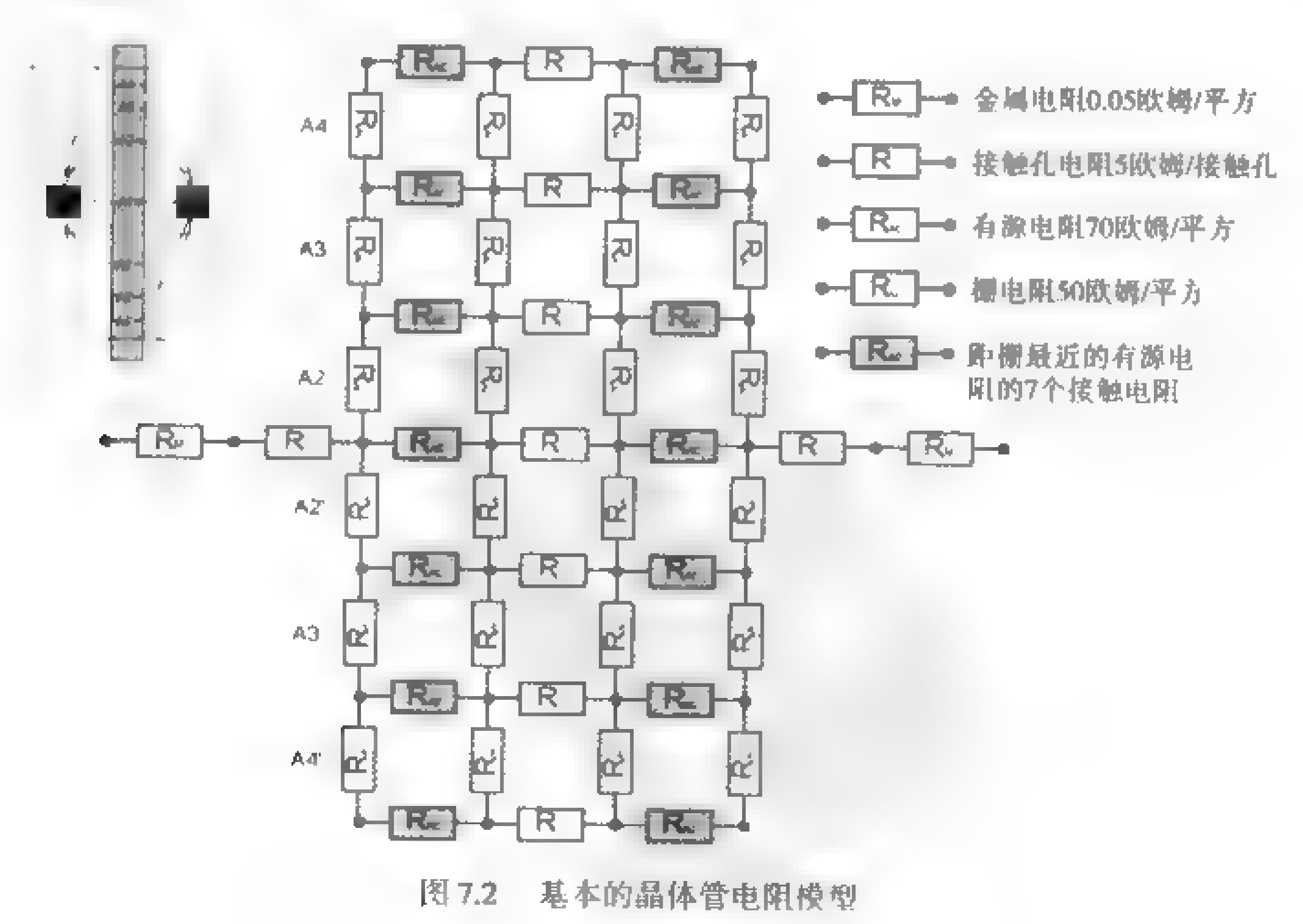


图7.2　基本的晶体管电阻模型

有趣的是，有源区的阻值是主要的，是金属1电阻的1000多倍，是金属1接触电阻的10多倍。这些数值对于最小化两条金属线间的总电阻给了一个起点：尽量最小化有源区的电阻！

为了解释这个概念，图7.3说明了在晶体管设计中不同接触方案的影响。

在表7.1中，对图7.3所指出的三个方案做出了比较。

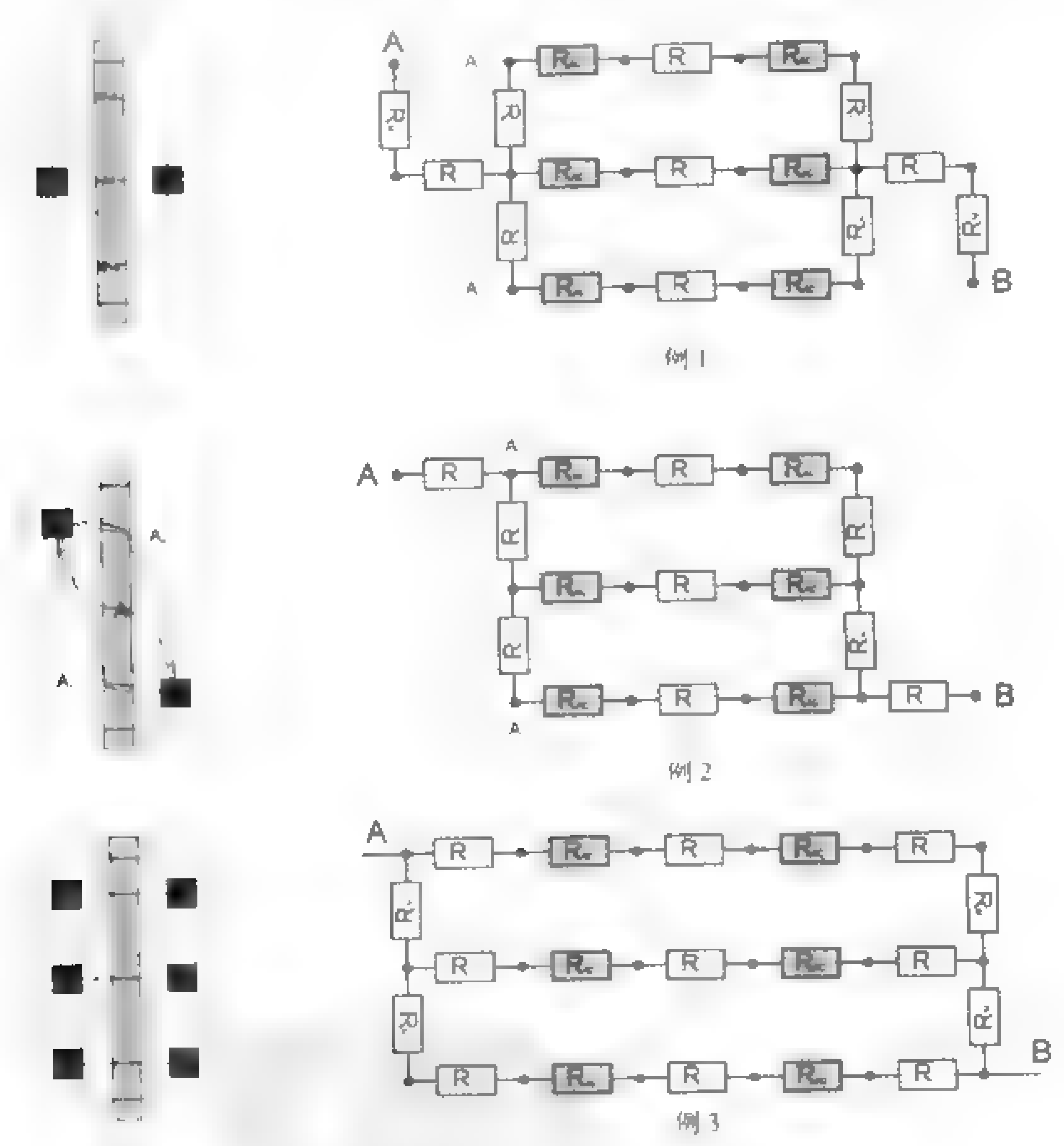


图7.3 晶体管接触方案的例子

所有这些例子在分析时都考虑了接触孔电阻。接触孔是金属或多晶的三维圆柱，它们增加了通道的电阻。在任何连接中，接触孔的数量很重要，需要加以考虑，因为对于每一个接触孔而言，通过从导体中引入另一平行电流路径，可以减少电阻。对于大电流的传输信号线如时钟和电源，这一点尤其重要。

对于晶体管来说，接触孔的版图设计方式和数目，应该依据具体的应用和设计中需要优先考虑的特性来决定。

表7.1　等效晶体管电阻的比较

例子	总等效电阻	注释
例1	56.1欧姆	基本版图
例2	105.1欧姆	由于有源层到接触孔的布线而增大的电阻
		在接触孔间的金属1布线沟道
		布线和性能之间的折中
例3	24.7欧姆	最小的电阻结构
		在金属1里可能没有布线
		由于有多个接触孔而产生了更好的可靠性
例4	450欧姆	最大的电阻结构
		在金属1里可能有四个布线沟道
		最小的水平面积

例如，模拟电路、高速电路、射频电路和DRAM电路，都是一些高速率性能和高可靠性的应用，因此，这种电路的标准元件是有充分接触的晶体管。如果可布线性是关键问题，那么选择图7.3的例2中晶体管的版图是合适的。在特定情况下，例2所示电路的高电阻可能不会造成不利的影响；因此，可布线性的益处使得选择这种方式很有吸引力。

需要注意的是，计算得出的电阻值应当是彼此之间相对使用，而且这些电阻可以帮助我们估计这些连接类型间的差别。

图7.4所示的晶体管版图是一个极端的例子，虽然它说明了一种解决版图设计问题的方案，却牺牲了晶体管的性能来满足可布线性的要求。有些情况下这个版图是合适的。某些工艺在各层电阻率特性方面差异显著。举个例子，有些工艺里，有源层用金属处理，这样得到的层的电阻率比我们例子里的电阻率更低（1/10）。在这个例子中，晶体管的性能是可以接受的，而且我们能够利用布线通道来优化单元版图。这些特殊的工艺比通常的工艺更加昂贵，但是我们应该时刻注意利用这些特殊的特性来进行版图设计。

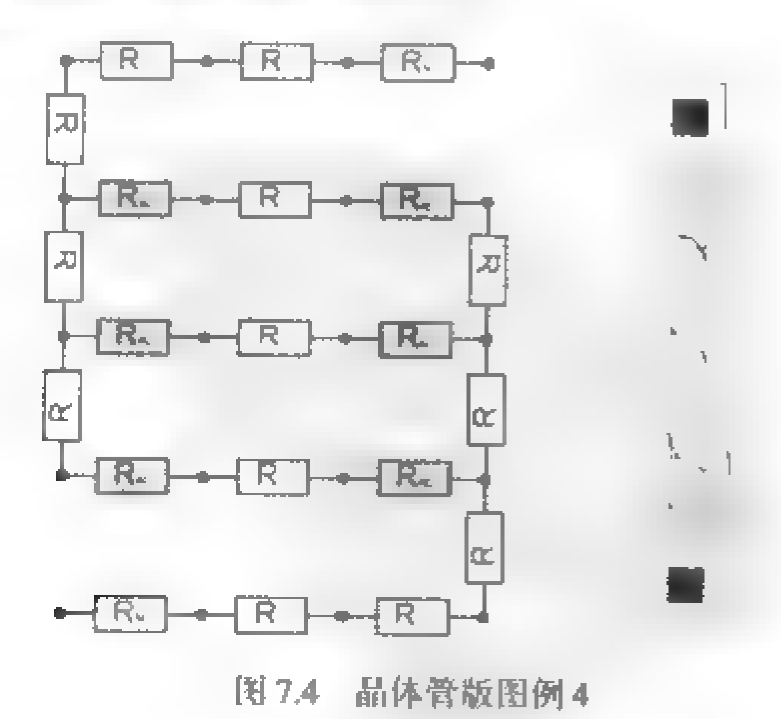


图7.4　晶体管版图例4

7.1.2 设计电阻

对版图设计工程师来说，控制电阻是一个需要理解的重点概念。在有些设计中，需要有目的地增加电阻。

需要电阻的情况如下：

- 分压器
- 延迟元件
- 动态逻辑负载
- SRAM 单元
- ESD 输入保护结构
- 很多模拟电路应用

在所有这些例子中，我们不但需要一个电阻，更需要一个阻值精确的电阻。

实现任何电阻的第一步都是选择一个恰当的分层。分层的恰当选择，可能需要考虑以下几个因素：

- 不同分层的电阻率。
- 在不同工艺和不同环境条件（温度）下电阻率的变化。
- 在不同工艺条件下分层的宽度的变化。
- 在所选择的分层上电阻的最终实现面积。

在大多数设计中，多晶栅被选择作为电阻的材料，因为这种材料的电阻相对较大，电阻率和宽度被严格控制，而且最终电阻所占的面积没有限制。有些工艺有一个专门的高电阻层，用这个分层来实现电阻是很理想的。

我们使用平方面积计算电阻，因此为了实现一个确定的电阻值，我们必须首先选择一个固定的宽，然后利用如下的公式来计算所需要的多晶的长：

$$R = \rho \times l/w$$

对于一个确定的电阻，较宽的多晶将产生一个更长的路径，因此多晶的宽度应当慎重选择，以便实现的电阻尺寸合理。通常，我们不选择最小设计规则的多晶，因为在这种条件下，宽度的变化是最明显的。在整个芯片中，将作为电阻的多晶的宽度标准化是一个好的设计思想，那样可以使得芯片上所有电阻的变化相同。一定不要忘记考虑接触孔电阻的影响！

图 7.5 所示为一个典型的多晶电阻的例子。这些电阻是用栅多晶组成的，通过接触孔连接到信号，用一个专门的电阻层来定义。这一层仅做记录和 LVS 之用。这一分层确定电阻存在的区域，在那里为了版图验证而有意识地用电阻去建立器件。CAD 工具要求接触层不覆盖电阻鉴别层。

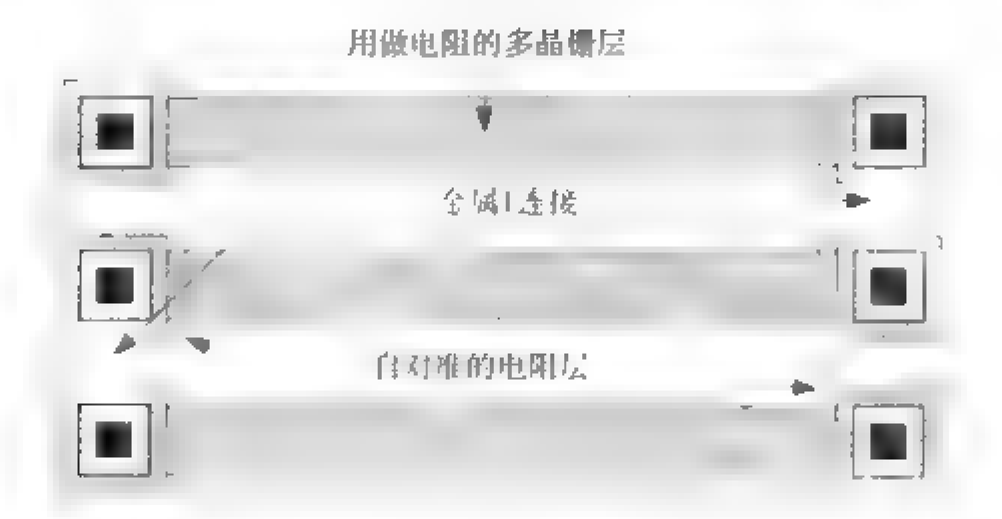


图7.5　典型的电阻

注意，电阻主要通过金属连接，因为它们主要使用在模拟电路中，而模拟电路常常需要信号微调。

从图7.6可以看出，这些是更加有效的面积解决方案，因为它们有时必须满足晶体管的面积需求，同时又不能占据太多的空间。但是这些电阻的缺点是，由于多晶层存在拐角，使电阻的计算变得不那么容易。作为最佳的近似值，在计算多晶总电阻时，我们可以使用宽度中心线的长来计算总电阻。

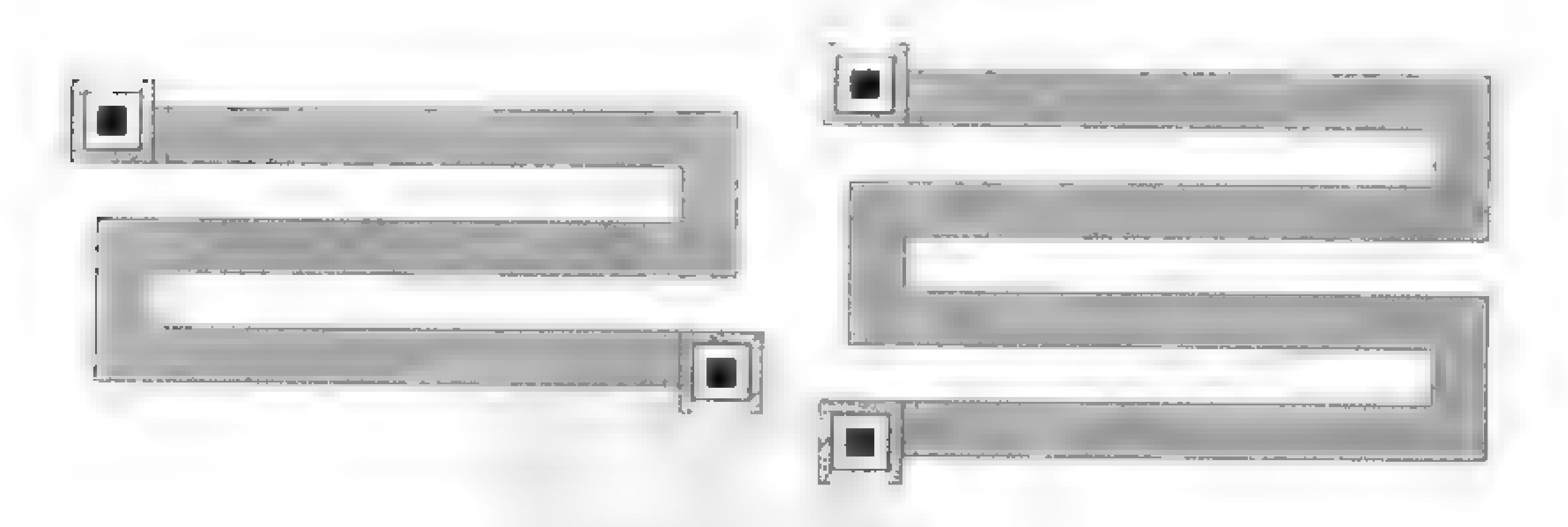


图7.6　有效的电阻类型

在模拟和射频电路设计中，我们将指状电阻屏蔽起来以避免耦合。图7.7展示的就是这样一个电阻。

在一些非常敏感的电路中，当两个电阻分别连接到周期相同、相位相反的信号时，为了确保两个电阻都按相似的方式工作，我们能够均衡两个电阻间的耦合效应。这样的电阻叫做平衡交错电阻，如图7.8所示的例子。

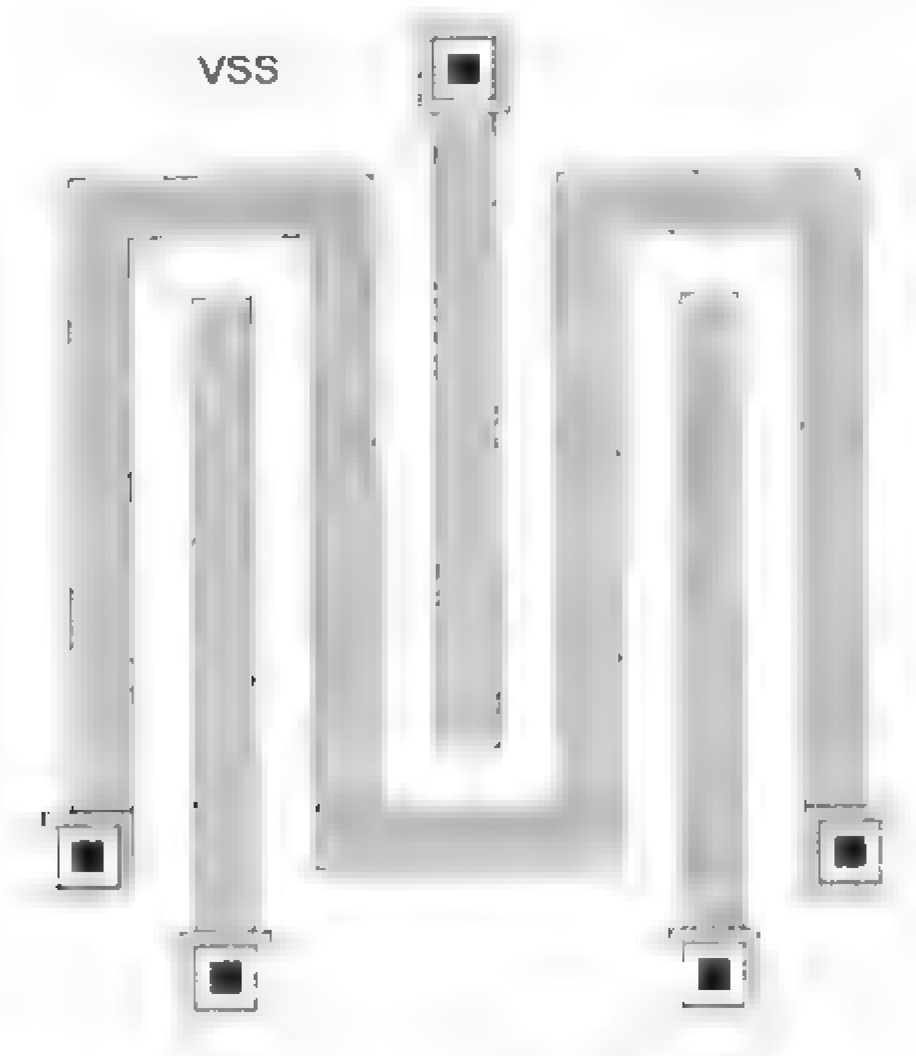


图 7.7 屏蔽电阻

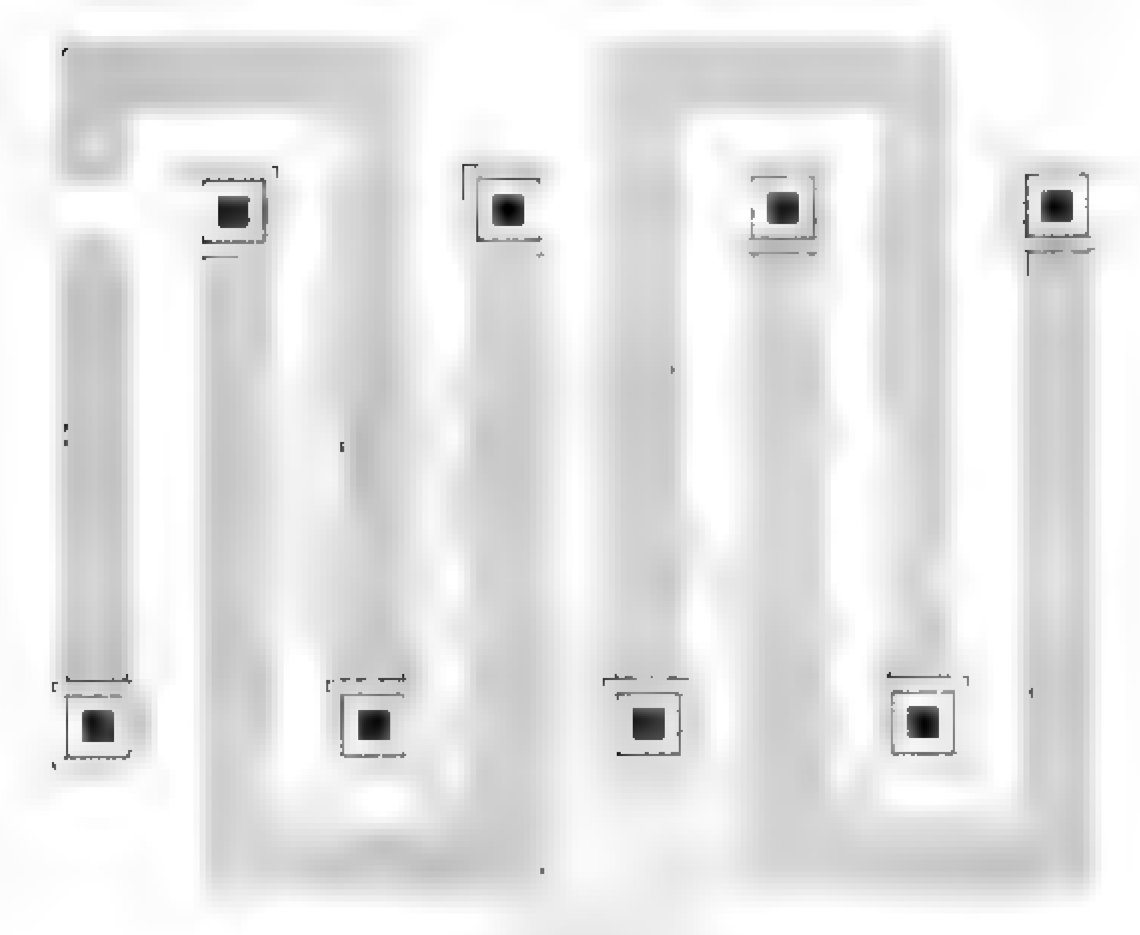


图 7.8 平衡交错电阻

7.2　电容

对版图设计工程师来说，电容的定义和来源是需要理解的重要概念。在一些特殊的设计中，电路原理图上需要有电容，但是，通常优化一个版图设计的重点是使不同版图结构中的固有的寄生电容最小化。

电容影响版图设计中的几种不同特性。当两个等效的设计放在一起比较时，电容较大的设计会导致下列参数值增加：

- 信号延迟
- 功耗
- 与附近结构间的耦合作用

回顾电容定义将帮助我们理解如下问题，即在各种各样的版图设计风格中，优良的设计可以减小电容。

通常用如下公式计算一个导体的电容：

$$C = \varepsilon \times A/d$$

在这里，A表示确定的导体的表面积，d表示导体和参考点间的物理距离，ε是一个常量，表示导体和参考点间的绝缘层的特性。图7.9所示为一个电容理论上的定义。

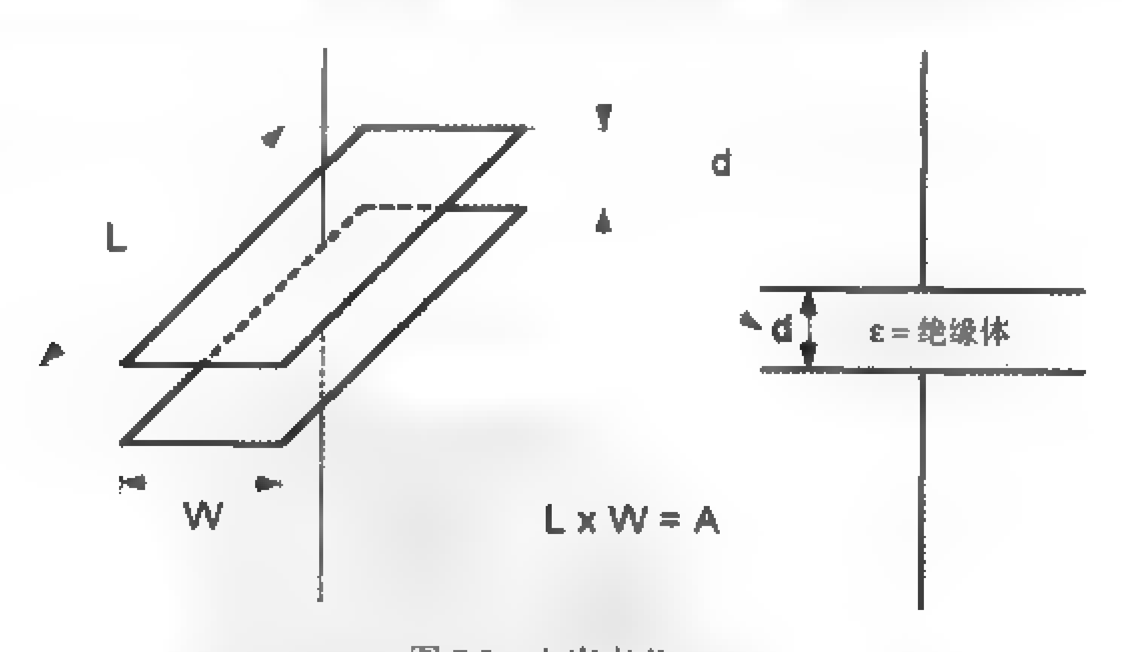


图7.9　电容定义

从公式中可以看出，显然有两种使信号电容最小化的方法（ε是制造工艺的一个特性，不能由版图设计工程师控制）：

1. 减小电容的面积——这个方法意味着减小了“极板”或者多晶之间的重叠区域
2. 增加电容的两个极板间的距离。

对一个信号来说，电容 C 限制了线上电压转换率（dV/dt），由如下公式可知，dV/dt 需要更多的电荷或者电流 I：

$$dV/dt = I/C$$

从这个关系，我们可以推导出延迟公式 $t_{delay} = RC$。

7.2.1 设计电容器

如前所述，在有些情况下，电容器是作为电路设计中的一个完整部分存在的。如下面的几个例子：

- DRAM 存储单元。
- 电源去耦电容器。
- 电源发生器的充电电容器。
- 延迟链。
- 特殊的模拟电路，如开关电容器应用。

怎样才能使电容器按预期的设计来实现呢？

在第 5 章中曾讲到，一个 DRAM 存储单元电容器是两个已经过优化的非常靠近的分层的简单交迭。在这种情况下，距离项 d 已被最小化，以增大电容值。

在一个标准CMOS工艺中，选择分层来实现电容器实际上只限于一种情况：根据设计，制造一个晶体管，要求栅多晶层和有源层的距离很小。源和漏的节点连接到电容器的一个终端上，栅的节点连接到另外一个终端上。

图 7.10 例示了两个基于一个 NMOS 晶体管的电容器的实现。一个在衬底，另一个在 N 阱中。N 阱中的晶体管产生了更大的有效电容，因为 N 阱中的 MOS 管有更低的开启电压。

电源去耦电容器可能是最常用的电容器，因此有必要对这些电容器做一些解释。这些电容器连接在电源的两个节点间（例如 VDD 和 VSS 间），用来耦合两个节点，并且给产生大的噪声的信号提供动态电荷。这些去耦电容器用来稳定电源电压，同时增加芯片工作的可靠性。

为了产生数量可测的电荷，这些电容器需要遍布在芯片中，并且总电容的有效大小应该在纳法（nF）的范围内。为了有效地实现，需要做一些规划；但是，在很多场合中，电源线通常是很容易得到的，因此，规则的电路一旦实现，在版图上找到剩余空间不是很困难。我们建议，在一些不需要去耦合电容器的电路中，用保护环来隔离这些大的晶体管，以避免噪声耦合。

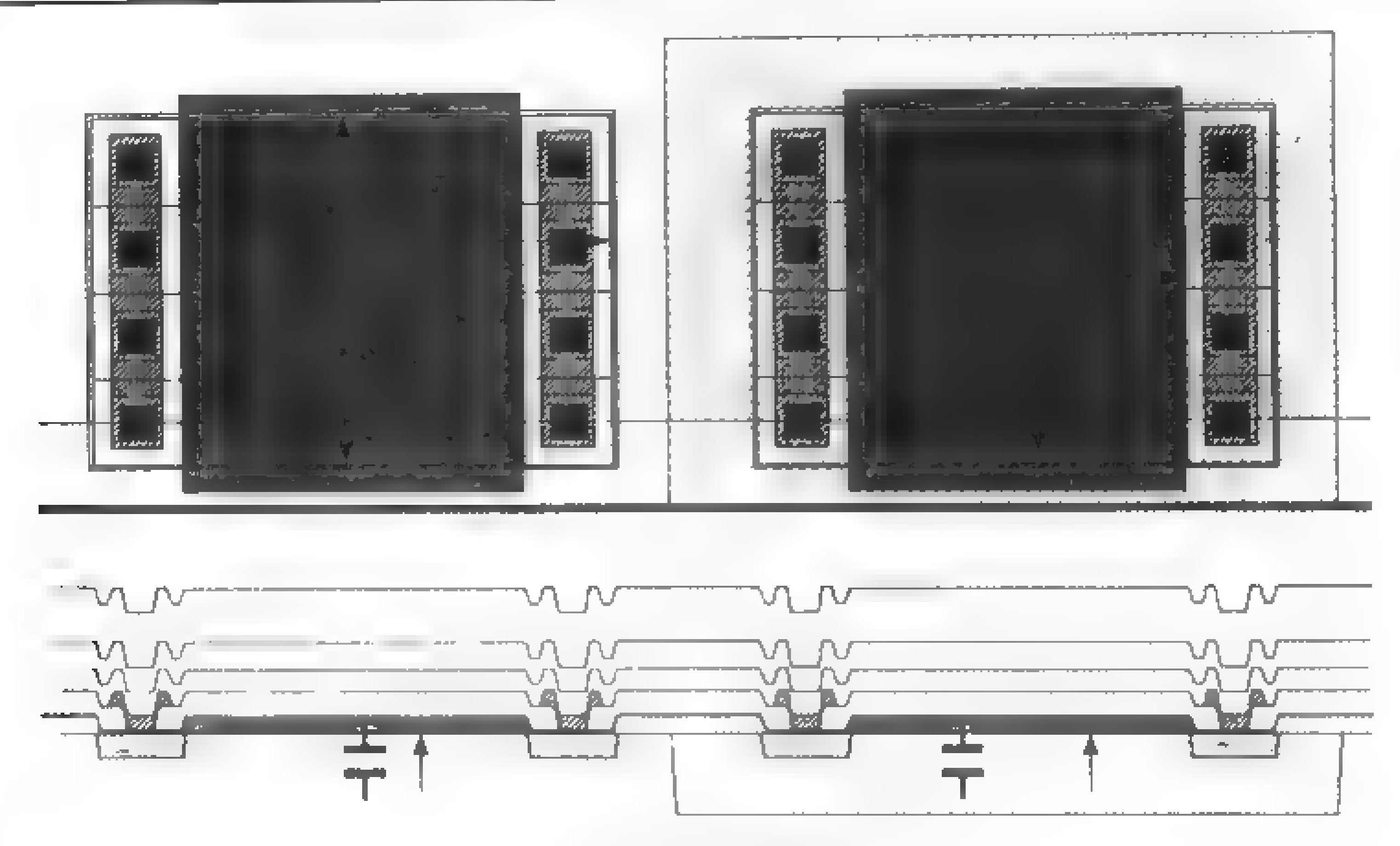


图7.10 晶体管电容器

7.2.2 最小化晶体管寄生电容

如图7.11所示，在晶体管中存在许多固有寄生电容

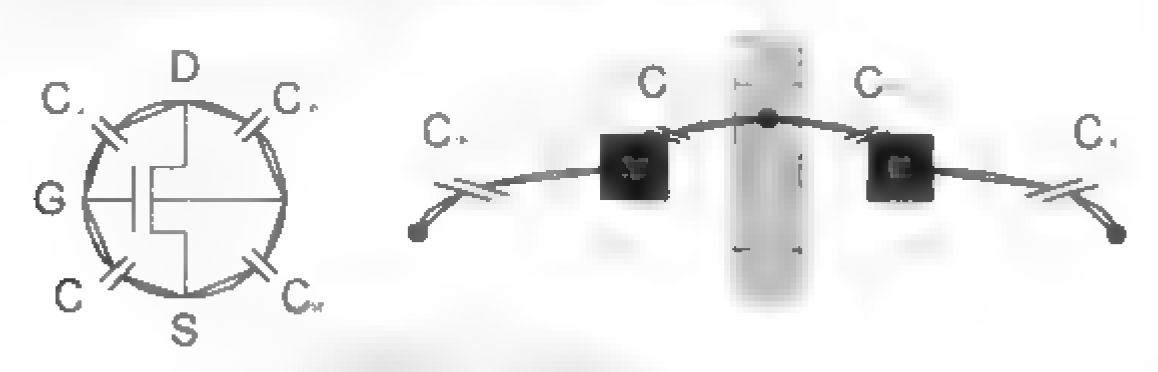


图7.11 晶体管电容模型

图7.12指出了一些晶体管设计的例子，这些设计可以明显减小漏电容C_{DB} 表7.2给出了对图7.12的解释

通常，如果在设计中使用最小设计规则，那么可以使用由图7.12所示的设计方法，来优化晶体管的电容

7.2.3 互连电容

在设计中，寄生的互连电容无法避免，并且在大多数情况下，寄生互连电容对电路是有影响的，必须通过对电路设计的负载进行合适的补偿来处理这种影响

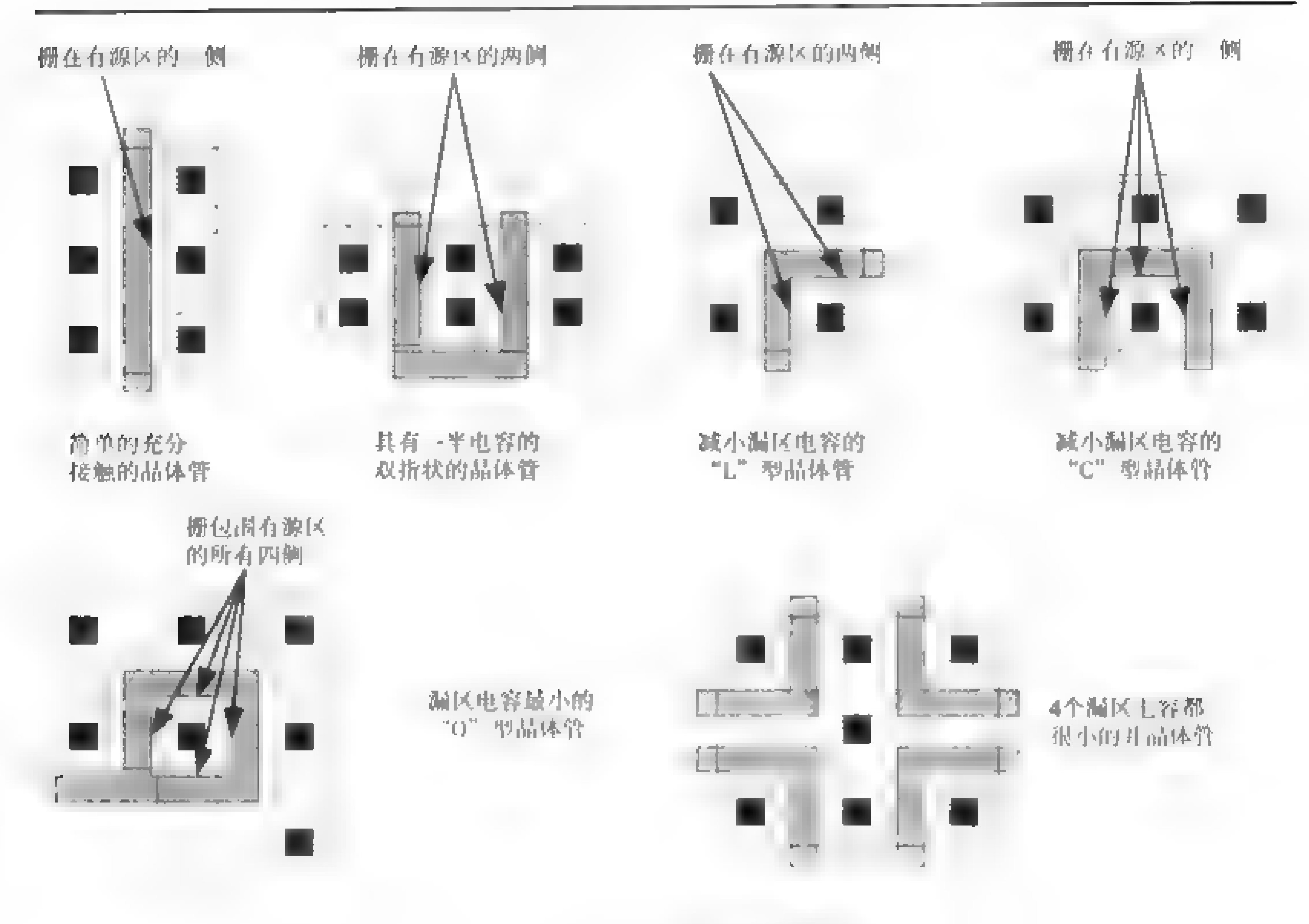


图 7.12　最小电容晶体管

表 7.2　图 7.12 的解释

例子	解释
简单型	● 标准版图
两个指状型	● 与标准版图比较，C_{DB} 至少减半
	● 偶数个指状晶体管的最佳使用
"L" 型	● 小的漏区，对于器件尺寸在一个接触孔大小左右的较好
"C" 型	● 小的漏区，更容易连接栅的两端
	● 可以用在数据通道或者总线接口应用中
	● 绝对最小化漏电容，适于数据通道的应用
"O" 型	● 基于"巧克力"或"华夫饼干"晶体管；更多的例子参见本书附带光盘
	● 对非常大的晶体管来说，电迁移规则限制了此晶体管的使用
Flower 型（#字型）	● "L" 型的扩展，用于 4 个输出节点

在设计中，提取工具主要用来建模和计算每个节点的互连电容。在介绍了电容的来源后，本节将介绍用提取工具模拟和计算电容的概念。

鉴于芯片是通过层层覆盖、多层叠加的方式制造出来的，在一个特定的多晶附近有很多实体，每一对近体都可形成一个寄生电容。在现代工艺中，一般认为任何特定节点的总电容的70%到80%是由互连布线的寄生电容产生的。由于模型尺寸的增大，互连层数量的增加，以及行距的减小，处理的办法也由晶体管负载转向了互连负载。

一旦理解了互连电容的来源，那么现在要做的就是用先进的技术处理互连电容的问题。图7.13是一个针对不同情况的三维视图，用来解释互连电容的不同来源。

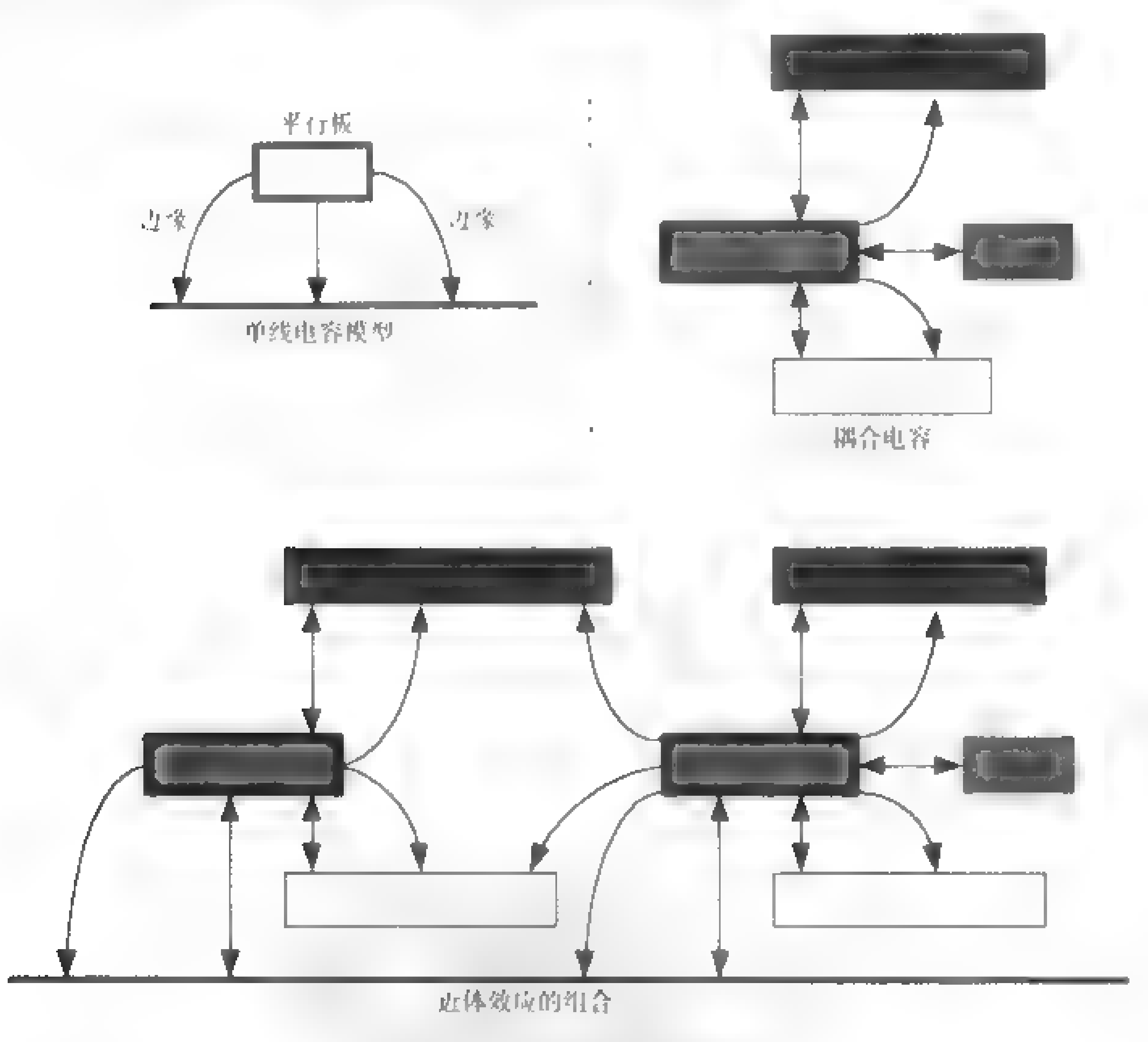


图7.13 互连布线电容的例子

在本书附带光盘中有一个提取的实际例子。提取就是试图把电路的真实数据反馈到电路设计中。

如图7.13单线电容模型部分所示，任何信号线的电容基本上由三种不同成分组成。

1. **平行板电容**：即在7.2节开始部分所描述的简单电容模型。
2. **边缘电容**：由电流流过线时所产生的感应电场引起的。

电容的模拟和计算不在本书所讲解的范围内，但是边缘电容在总的互连电容中占很高的比重（~50%），理解这一点很重要。

边缘电容的大小依赖于信号线与所需考虑的实体的距离 d，只要谈到这点就足够了。

直到最近，提取工具已经被开发来处理这个问题，在以前，提取工具仅仅局限于用来解决平行板电容类型的模型。

3. **耦合电容**：从一个信号节点到另外一个信号节点间的电容。

用简单的模型来模拟耦合电容，等于用平行板电容和边缘电容计算处于所需考虑的两个信号间的耦合电容。

我们在图7.13的耦合电容部分来尝试解释这个问题，如图所示，近体电容可以是信号线之间的或与其他结构之间的，例如电源节点和衬底。

精确的耦合电容模型仅仅适用于一些特殊场合，因为指望那些在仿真和提取网络尺寸中用到的工具和方法学能够考虑所有近体电容是不实际的。

一个节点的电容总是相对于地，这一假设，未免过于简单。当一个电容器两极板上的信号电压在同一时刻均发生相反的变化时，考虑耦合电容就显得至关重要。图7.14所示即为这种情况。

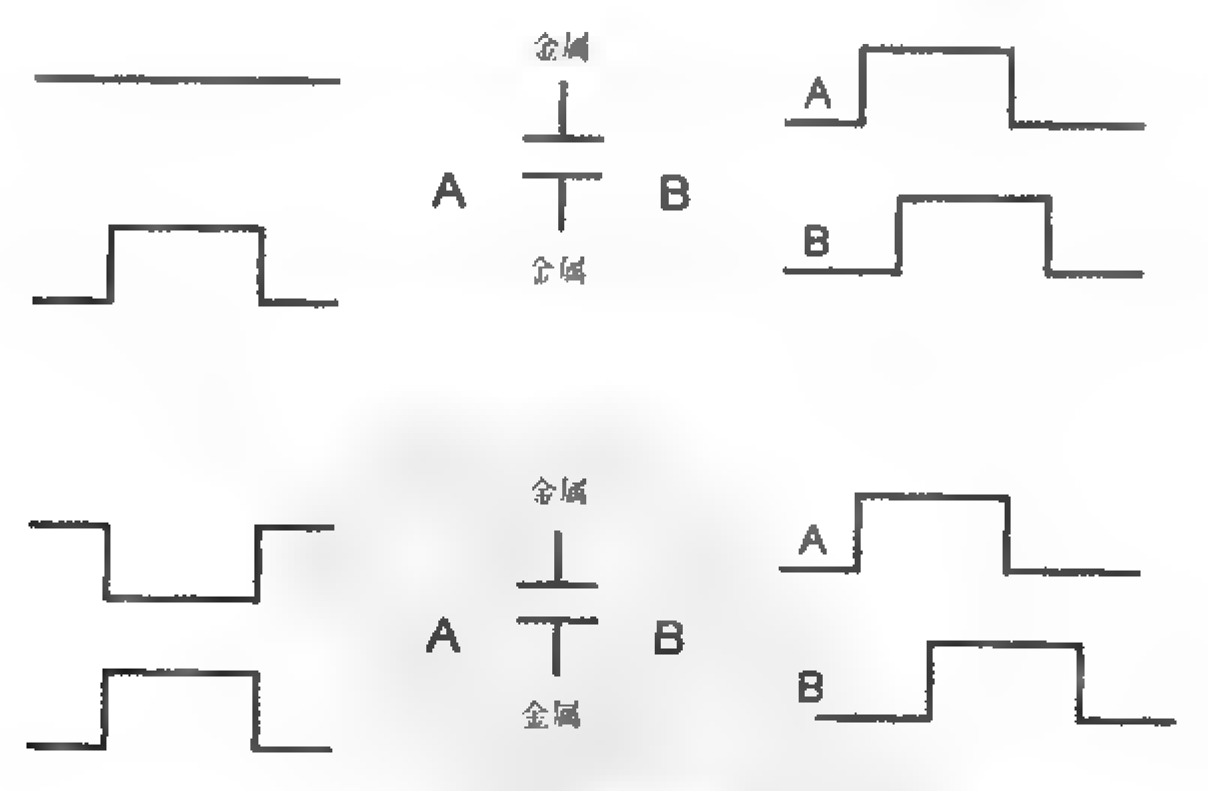


图7.14　耦合电容效应

在这种情况下，耦合电容对信号延迟的影响两倍于仅有一个信号的电压发生变化的情况，后者中参考点是一个静止的信号线的驱动器试着通过某种方式驱动信号，并同线的寄生电容

竞争。当电容器参考节点的电压朝相反的方向改变时，寄生电容耦合进了线，其结果就是增加了信号延迟。

与此相反的是，如果两个信号线都向同一个方向改变，那么它们互相帮助。

既然我们已经理解了寄生电容的来源，那么我们可以采取哪些方法来减小加载在一个已知信号上的各种电容负载呢？

- 减小平行板的面积。

  从版图的意义上来说，这意味着什么呢？减少信号线的长，或者最小化信号线的宽，或者长和宽都减小。

  需要指出的是，如果仅仅简单地把信号线宽减少50%，那么电容将仅仅减少25%，这是由于线的平行板部分仅仅占线的总数的50%，同时长度没有改变，边缘条件也保持固定。

  与此相对应的是，将给定信号的长度减小50%，将使电容减小50%，因为两个平行板和边缘条件也受到同等的影响。
- 减小距离 d 或者绝缘体的两平行板之间的距离。

  再一次，从版图的意义上来说，这又意味这什么呢？不管任何时候，可能都将关键的信号线布置在空的沟道上，同时使信号布线层或其他层上的面积最小化。

  例如，假定使用了一个空的布线沟道，同其他的布线层相比，使用最上面的布线层，相对于衬底的电容将是最少的。

  使用最上面的布线层实现，而且近似图7.13中所展示的单线电容模型那样的情况，是力求达到的理想状况。
- 为了处理两个信号的耦合电容，可以增加在同一层上的两个信号线的间距。我们称这个为信号散开，在布线拥塞很低时，这个方法十分有用。关于这部分的内容，请参见本书附带光盘中Sagantec的演示。
- 在差分信号的情况下，使用一个“交叉”方案，通过确保任何耦合都等价的影响两个信号，这个模式减少了邻近线间的耦合效应。图7.15用信号XX示范了这个概念，并引入了屏蔽的概念。
- 用一个处在静态水平的信号保护关键信号。这种方法消除了图7.14中解释的出现更糟糕的耦合情况的可能性。

屏蔽信号的概念是从印刷电路板（PCB）引进的。在印刷电路板中，一些用来给电路提供参考电压的信号，相比较其他信号而言，要求更大程度地与干扰隔离。图7.15展示了一个

屏蔽的例子，但是，屏蔽中还包括其他需要更详细叙述的技术。一个信号能够在两侧屏蔽，如图7.16所示。在这种情况下，我们隔离了信号，使之不受同一层的影响。

图7.17和图7.18展示了更加完善的屏蔽的例子，因为被讨论的信号完全被包围，包括信号线上侧和下侧的各个方向上。正如两个图所指出的，屏蔽的性能依赖于可利用的布线

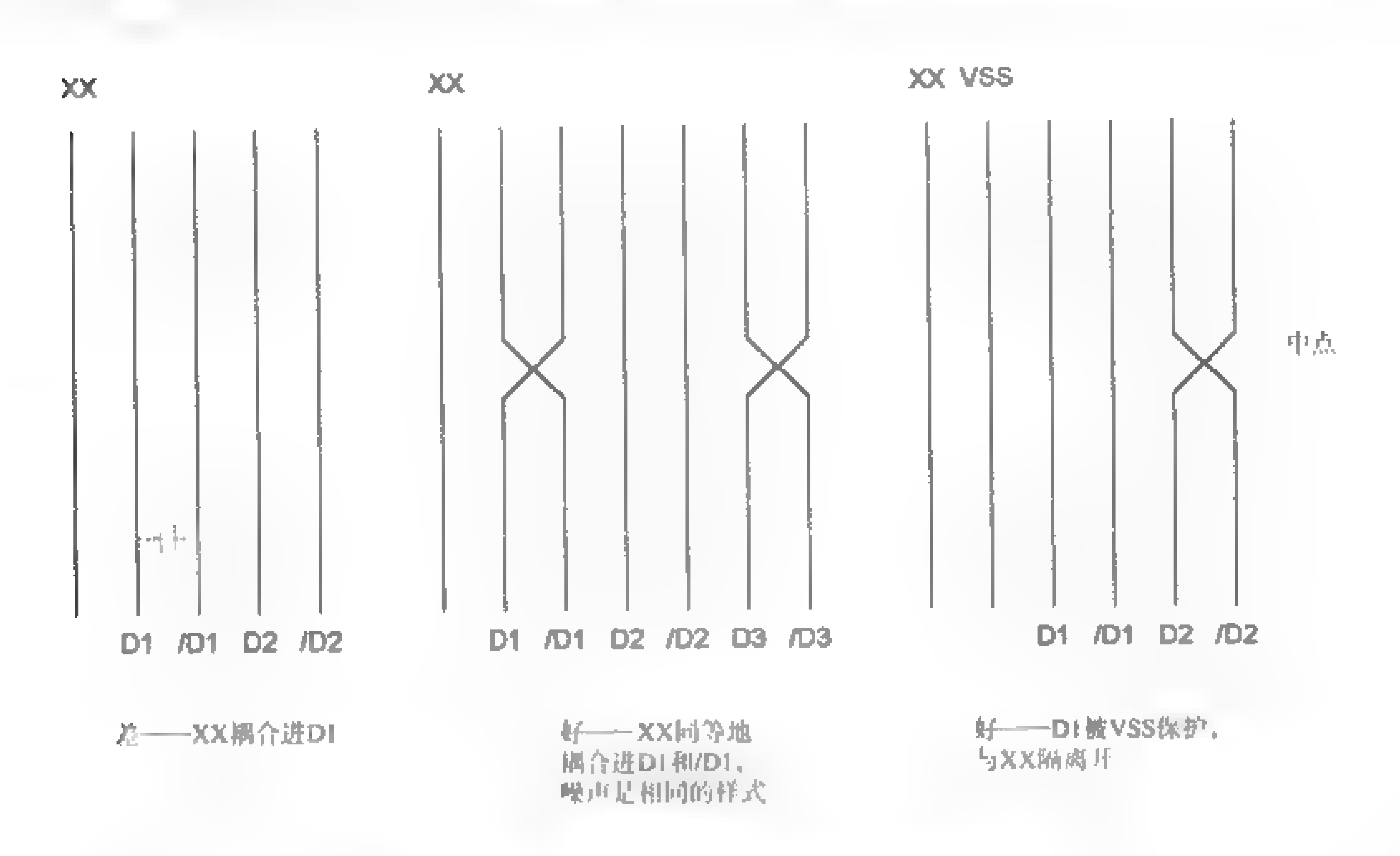


图7.15 差分对的旋转，用以降低信号耦合

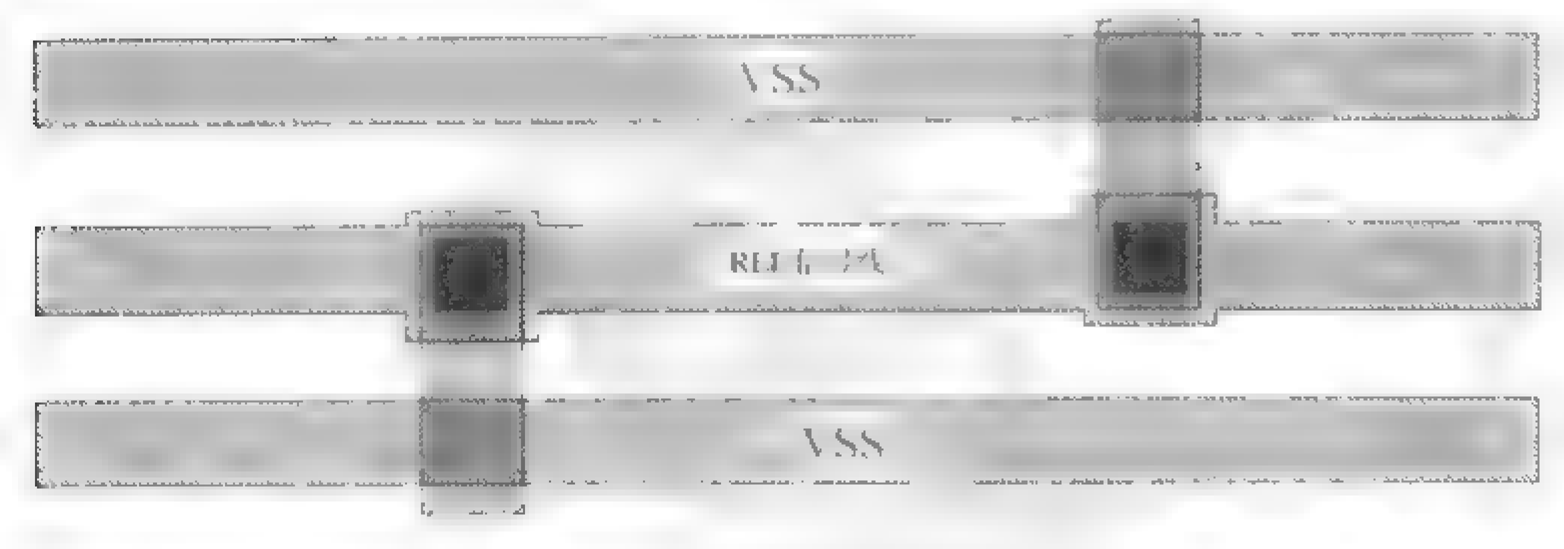
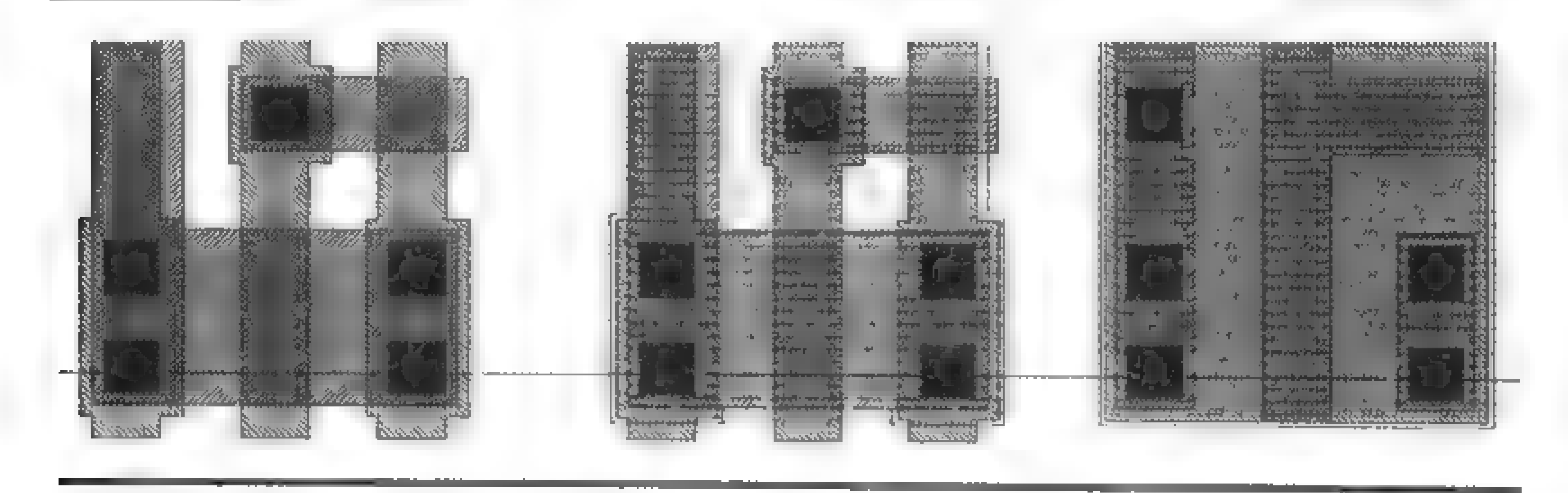


图7.16 简单屏蔽示例



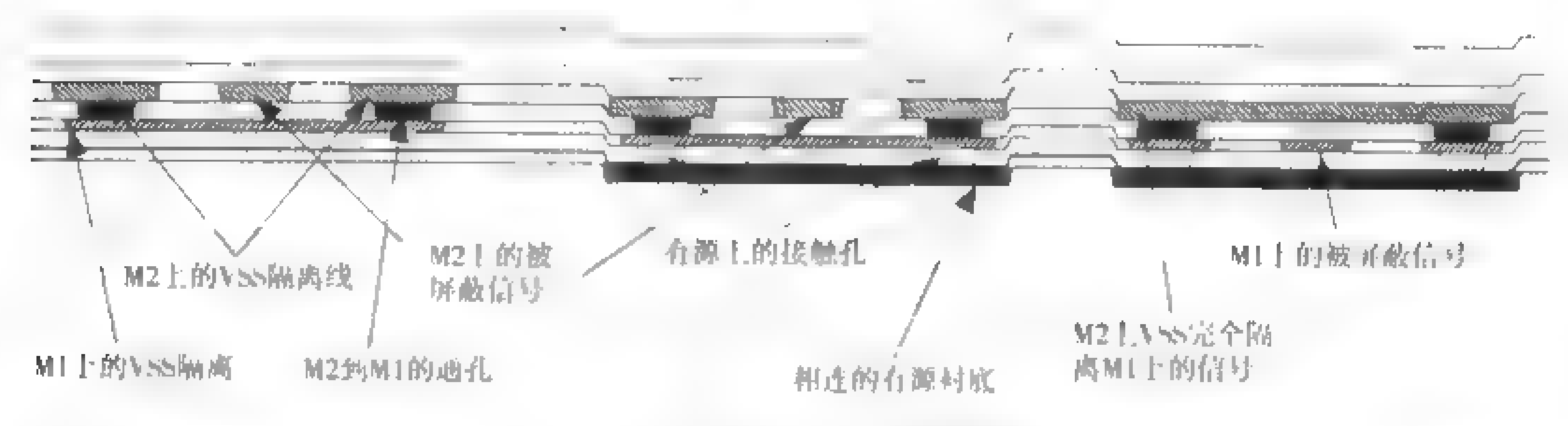


图7.17　两层金属工艺中的屏蔽选择

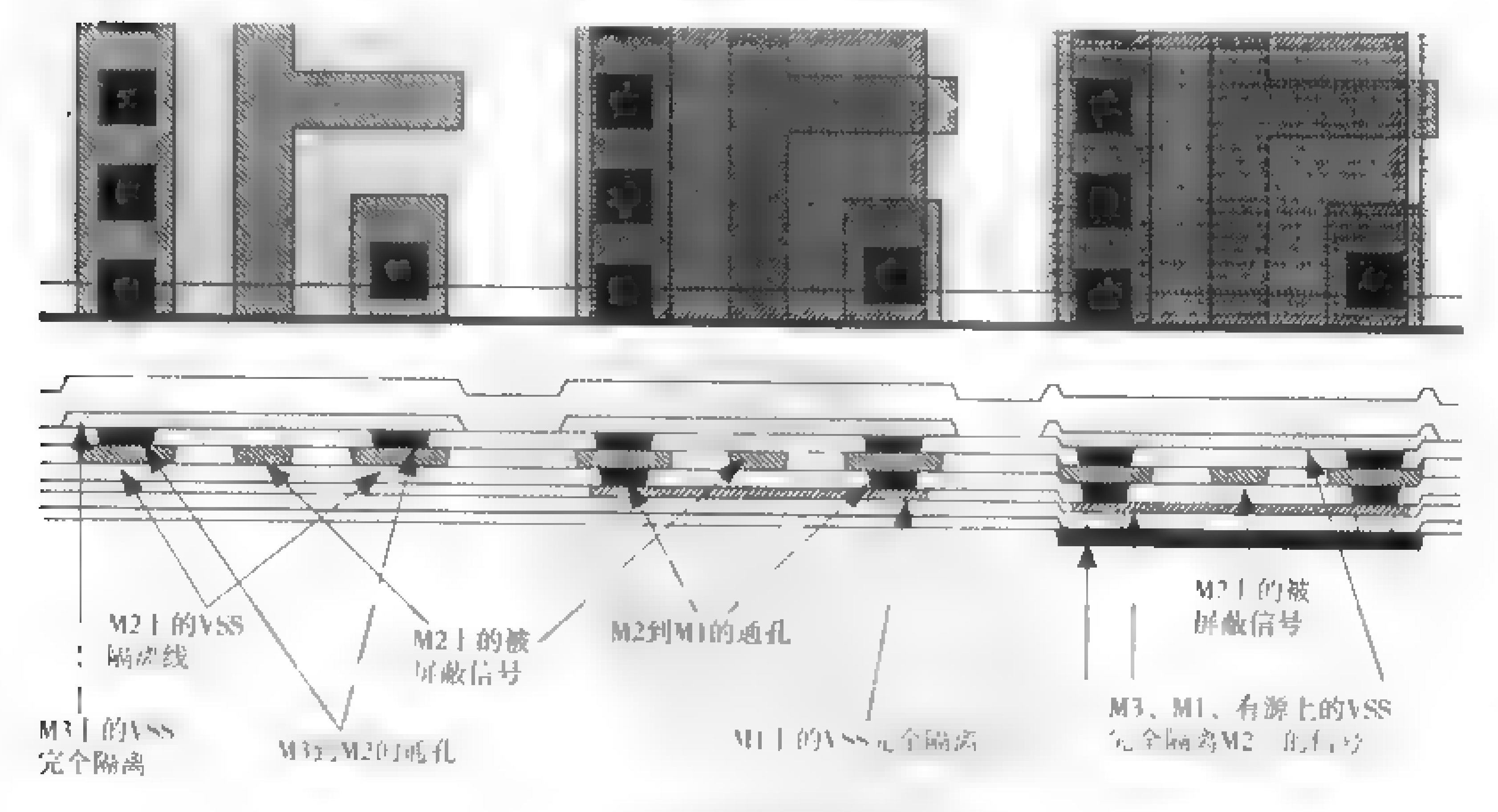


图7.18　三层金属工艺中的屏蔽选择

7.3 对称

设计性能的可预见性可以从很多方面来考虑。例如，我们在衡量一个电路的时序特性时，希望其能完全满足性能的要求。通常，我们要求两个版图的设计保持一致，这样两个电路的性能特性可以彼此相互匹配。

下面是我们经常使用的对称结构的一些例子：

- 差分放大器的设计，要求单元版图的两半相互匹配。
- 数据通道和存储阵列电路的版图设计，要求每一行和每一列的版图一致。
- 在具有相似结构的版图设计中，如多输入与非门，通常被建成一个单元；而在有一定差别的平行通道中，通常改变一些连接会消除互连元件的不对称。
- 在两个信号间，有差异的成对的布线需要具有匹配性。
- 设计能满足触发器跳转和保持时序要求的信号线和时钟线，是使信号线和时钟管线相匹配的一个实例。

确保两个电路工作状态一致的最好方法就是在两个电路的版图设计中，使用同一个单元。通过设计两个具有相同长、宽、负载以及耦合环境的信号线，可以达到信号对称。我们将进一步讨论这些概念。注意，在特殊情况下，本节介绍的所有设计技巧可以组合起来，应用到对信号变化十分灵敏的电路设计中去。

7.3.1 对称的版图设计

在许多模拟电路、射频电路或者灵敏的数字设计中，一个特殊设计的两个对半部分不仅在电气性能上相等，而且要求每个部分在工作性能上相同。差分放大器就是一个例子。差分放大器的电路功能是分辨出两个输入信号之间的信号差异。

为了得到几乎完全对称的版图，一个非常简便的设计方法就是把一个版图单元重复使用两次。两个版图单元之间有较小差异的地方，应该在单元的顶层上进行操作修改。图7.19阐述了这种设计方法。

这种设计方法有其优点，也有其弊端，综述如下：

- 可以确保版图的对称，并且容易操作。
- 可以发挥基于单元的版图设计的优点。只需修改一个单元，就可以修改对称版图中的两个对半部分。
- 单元的设计过程有些复杂，因为使用这种方法首先要确定版图中哪些部分是基本对称的，在完成版图设计的过程中，必须始终考虑版图中的对称部分。
- 由于对称，可能会有一些区域或布线开销，但是沿着对称中心线上的多边形通常是共用的。

经验提示：使用“Edit in Place”模式来建立底层单元。这种模式是嵌入到所有的版图编辑器。只需在顶层中修改两个底层单元中的一个，就可同时修改另一个底层单元。

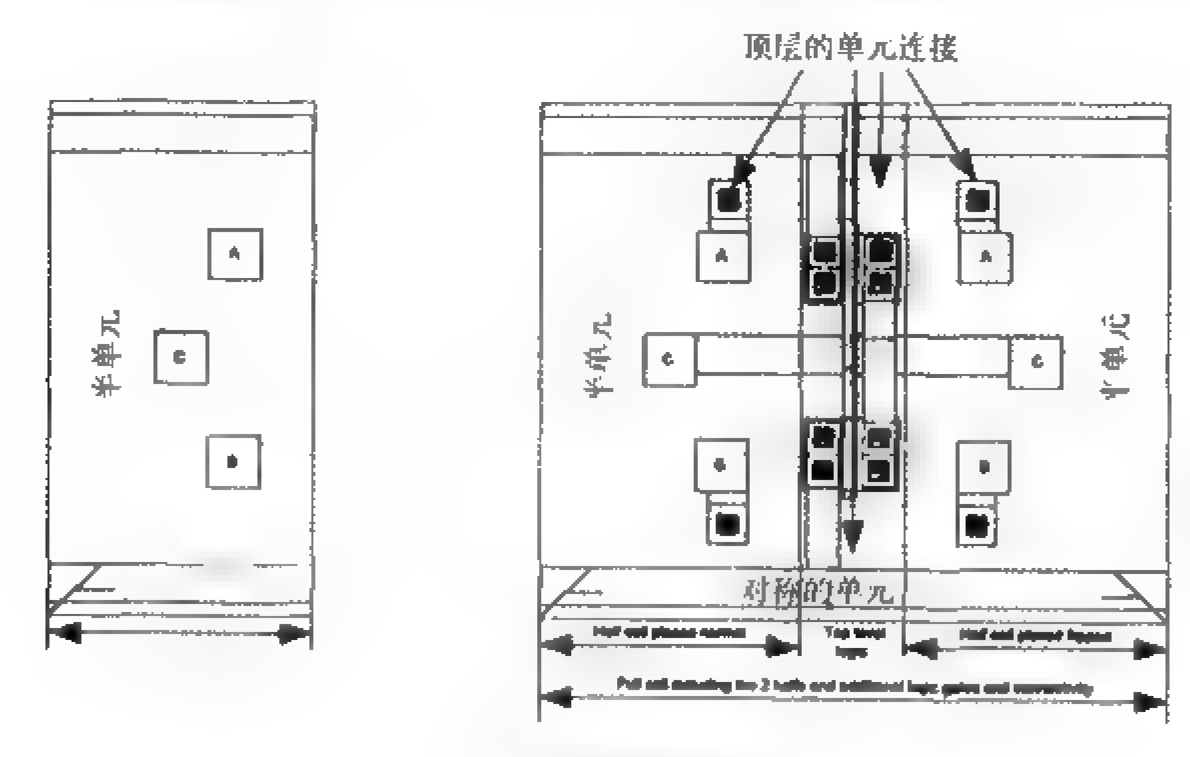


图7.19 对称版图的例子

7.3.2 平衡的版图设计

所谓平衡版图设计，就是通过在内部结构上的巧妙连接或者对称的版图来实现性能上的对称。

通常，设计对称化的电路会产生如下的结果：

- 减小功耗，设计人员都在想办法解决芯片的高功耗问题，并试图避免使用风扇来降低芯片温度，以确保其工作在最佳温度下。
- 时序对称。在模拟和RF电路设计中，这一点尤其重要，因为在这些电路中，每个转换器件的时序都非常关键。
- 为了准确得到设计的对称化特性，需要定义更详细的连接模型。

对称化的版图设计，是从一种被称为“对称器件”的老的模拟电路设计方法中衍生出来的。这种设计方法被应用到了IC设计中，图7.20所示的例子，用两输入与非门阐述了这个概念。采用这种设计，我们可以使信号分别从两个不同的输入端进入，再从输出端输出，而两个所经过的延时相等。

图7.20右边所示的电路图，准确地指出了第3个版图示例的电路图是如何定义的。许多LVS版图验证工具有可以从版图中辨别与非门的运算法则，但是，第3个版图示例所示的版图，通常不能被识别出是一个与非门。因为与常规的与非门电路图相比，第3个版图示例定义的电路图是有差别的。产生这些问题的原因是，在这个特殊的与非门中，两个串联连接的顺序被颠

倒了。功能上，这两个串联连接是等价的，事实上也是对称的。但在做LVS验证时，在这个特殊的与非门示例中，我们必须修改电路图，以使其通过LVS验证。

1. 简单与非门
2. 双指与非门
3. 对称与非门

图7.20 对称两输入与非门

平衡电路，并不总是像上面例子所示的那样直观明了。当遇到多于两个晶体管串联的情况时，对称串联器件就更难看出了。图7.21所示的三个串联门的例子，进一步阐述了这个概念。

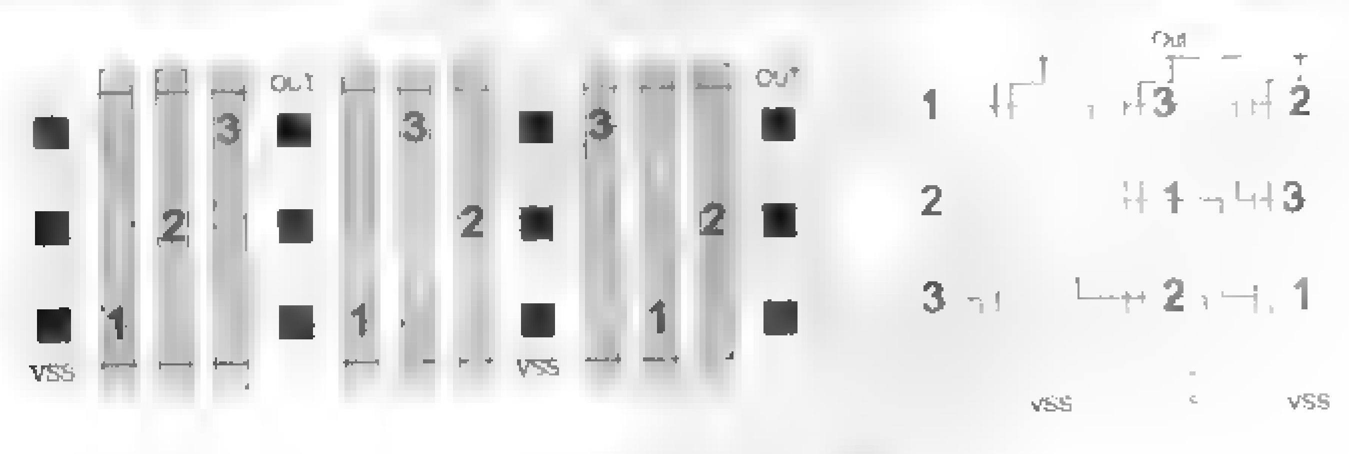


图7.21 对称的NMOS串联晶体管

为了平衡串联连接，每个输入端分别接在三个晶体管所在位置中的一个：分别为靠近输出端的、中间的和靠近电源的。这只在有三个平行的串联链时才是可能的；因此，在一个版图设计中引入平衡后，可能导致版图资源的显著开销。

7.3.3 物理补偿

我们使用物理补偿这个术语来描述这样一个概念，即把版图的对称化设计应用到信号线上。我们通过设计两个长、宽、负载以及匹配环境都相同的信号线，来达到信号对称。

设计信号线和时钟线来满足触发器建立和保持时间的要求，这是使信号线和时钟线相匹配的一个实例。在有些情况下，如果时序容限非常小，就需要使用物理补偿。

同步DRAM就是一个很好的应用示例，其中输入的建立和保持时间都需要精确的定义。因为在同步DRAM中，在各种电压、温度和工作状况下，电路性能都需要有所保证。这种情况下就需要应用物理补偿。

如前所述，我们可以比较信号线的各种结构特点，通过对这些结构特点的补偿，来实现信号对称。这些结构特点包括通孔的数量；布线层的匹配和每一层中线的长度；以及均等地屏蔽每条信号线。

下面所示的是对一组信号线实现物理补偿步骤的一个简单列表：

1. 对所有信号线，采取统一的布线规则，并为长度补偿和屏蔽线预留空间。使用同一线宽布线。
2. 确定要补偿的一组线中最长的线，并增加其他所有与之匹配的信号线的长度。我们可以适当地弯曲一些信号线，只要可以保证足够的屏蔽就行。
3. 如果使用了多层布线，那么在每层上，都要使每条信号线的连接线长度相匹配。没有必要将在同一位置进行布线的不同互连层沿着某一条通道放置，但是在每条信号线上使用相同数量的通孔是很重要的。

   为每条连线匹配相对应的晶体管负载或者扇出。这意味着，晶体管的负载线的尺寸相对于驱动线的尺寸的比率，对每条连线都应该相同。物理补偿用的晶体管负载线应该加到适当的信号线上，以满足扇出比。
4. 表7.3阐述了补偿负载线计算的概念。
5. 如果需要，可以运用版图参数提取工具来验证结果并调整版图。

表7.3 补偿负载计算的例子

信号线	驱动器尺寸	负载	扇出比	补偿负载
A	100/50	500/250	5	—
B	20/10	40/20	2	60/30

图7.22所示的是在实现物理补偿以前的一组被简单布线的信号线。注意其中的箭头，小箭头所指的上方区域，是为后面的物理补偿安排的。图右边的长箭头指出了总线的尺寸。

图7.23所示为一个经过物理补偿后的版图。注意总线的最终高度比以前高，这说明该物理补偿方案要求的改动太多（小箭头指出了不同点）。

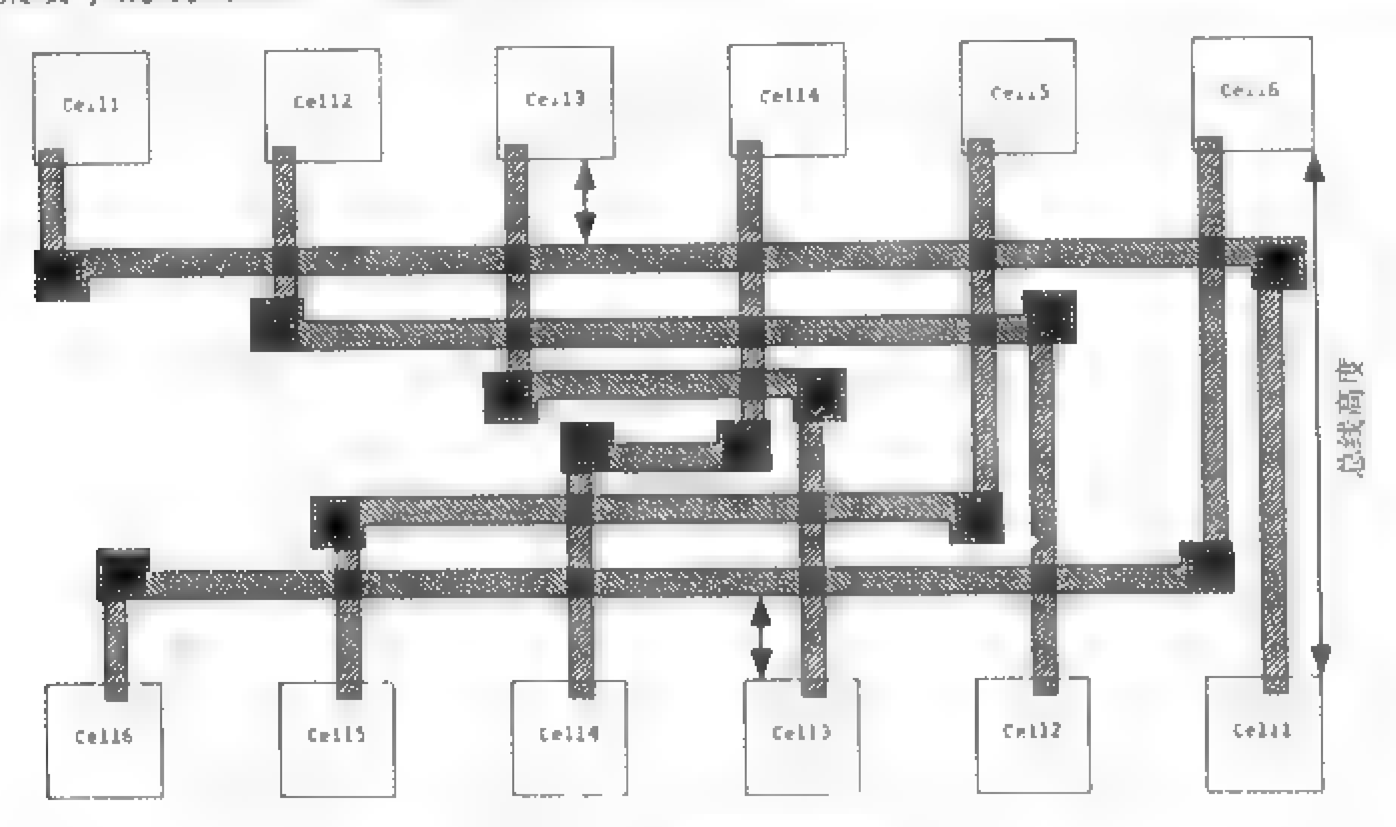


图7.22 简单的总线布线

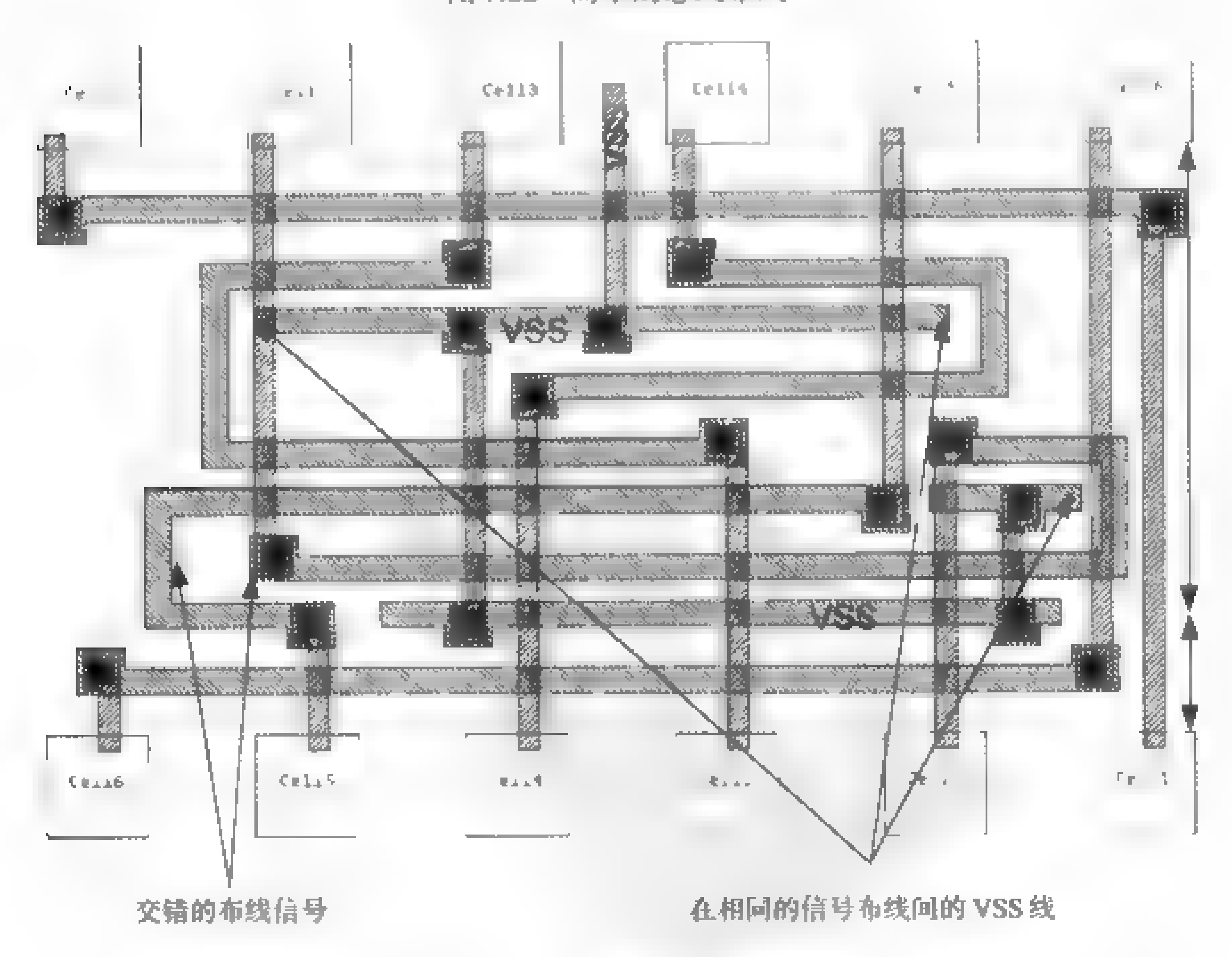


图7.23 物理补偿的例子

7.4　特殊的电气要求

7.4.1　版图中的 45° 角

我们已经解释过，对有 45° 角的多边形或通道信息的存储，会占用更多的数据库空间。因此，版图设计人员要尽量避免非 90° 角的多边形。然而，在某些情况下使用 45° 角，可能会减小版图面积或提高设计的可靠性。

含有 45° 角的多边形的电气参数，在特定的区域会很有用。当大电流流过导体时，90° 角拐角处所承受的压力很大。使用 45° 角多边形可以缓解这个问题。

以下情况，我们强烈推荐或者必须使用 45° 角版图。

- 在环绕拐角的地方，非常宽的电源线。所谓非常宽的标准是大于 30 μm 至 40 μm。在这种情况下，使用 45° 角会使大电流对于金属的压力得到缓解。
- 在相对拐角处的宽的连线或信号总线拐角。详见图 7.24。可以节省大量的面积。



图 7.24　宽的信号线的最优化布线

- 在靠近芯片拐角处布线的信号线，可以避免切割芯片时产生的压力。
- 实行 I/O 单元布局，旨在降低拐角处的电源振荡，在万一出现 ESD 情况时，可按前述将 I/O 单元排成锥状。
- 设计重复结构或间距受限的版图，以节省空间。
- 优化晶体管版图。使用 45° 角的版图的原因是很容易理解的，但是在设计晶体管时如何取得收益，却不是一件简单的事，所以有必要做进一步的解释。

在 7.2.2 节讲述最小化漏电容的内容中，我们已经见过栅弯曲晶体管了。这些晶体管不是最优的，带 90° 拐角区域的晶体管不能有效工作，因为这个区域不能流过大电流，所以 90° 拐角区域被认为是这个栅的死区。理论上，在计算晶体管的有效宽度时，栅弯曲晶体管拐角的宽度应忽略不计。

事实上，在这方面版图提取工具并不准确，因为它们使用的是一种简单的算法，这种算法用栅的中心线来测量栅的宽度。栅弯曲晶体管的栅的有效宽度按拐角的数量和栅长的比例应该有所减小。

图7.25表示45°角和90°角的晶体管的区别，实现了栅宽度的增加。

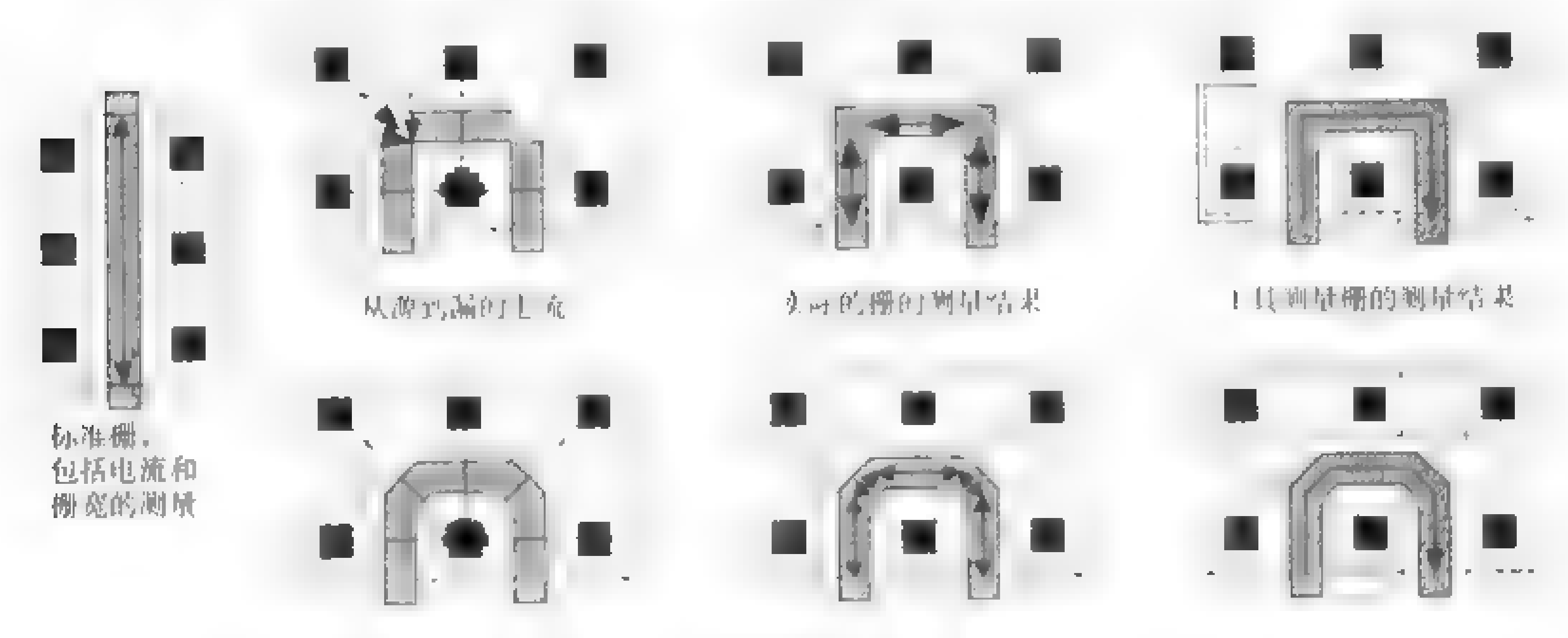


图7.25 栅弯曲的例子

在任何单元尺寸的增加都影响整个芯片尺寸的情况下，我们广泛使用45°角的栅。例如，弯曲栅，用于间距受限的单元，而这些单元在芯片中会多次用到。图7.26为一个字线驱动晶体管的例子，这个例子和很多PLA或压缩的寄存器组中的I/O单元很相似。使用45°角的多边形可以得到x和y方向上的最小尺寸的设计，但是这样的晶体管修改起来既费时又费事。

7.4.2 电迁移

电迁移指长时间的电流所导致的分子代替原子的现象，其结果是金属线最终断裂并形成开路。

设计拙劣的部分在经过一段正常的运行后，由于对电迁移的高度敏感而工作失常。随着集成电路尺寸的缩小，我们需要更多地关注由于电迁移带来的金属线的磁化系数，以及大电流密度和更高温度下电路的工作性能的降级。

电迁移现象在铝和铝合金互连的版图中是很普遍的。这就是在加工工艺中引入其他金属的原因之一。引入其他金属后，加工工艺不受电迁移的限制（例如，钨、铜）并可以得到非常低的电阻。

只需增加流过大电流的金属线的宽度，就可以避免电迁移造成的失败，这同样可以避免短路。因为电迁移依赖于导体中电流的大小，所以可以通过一系列的准则来指导版图设计人员有效地处理这个问题，而不会引起显著的面积开销。

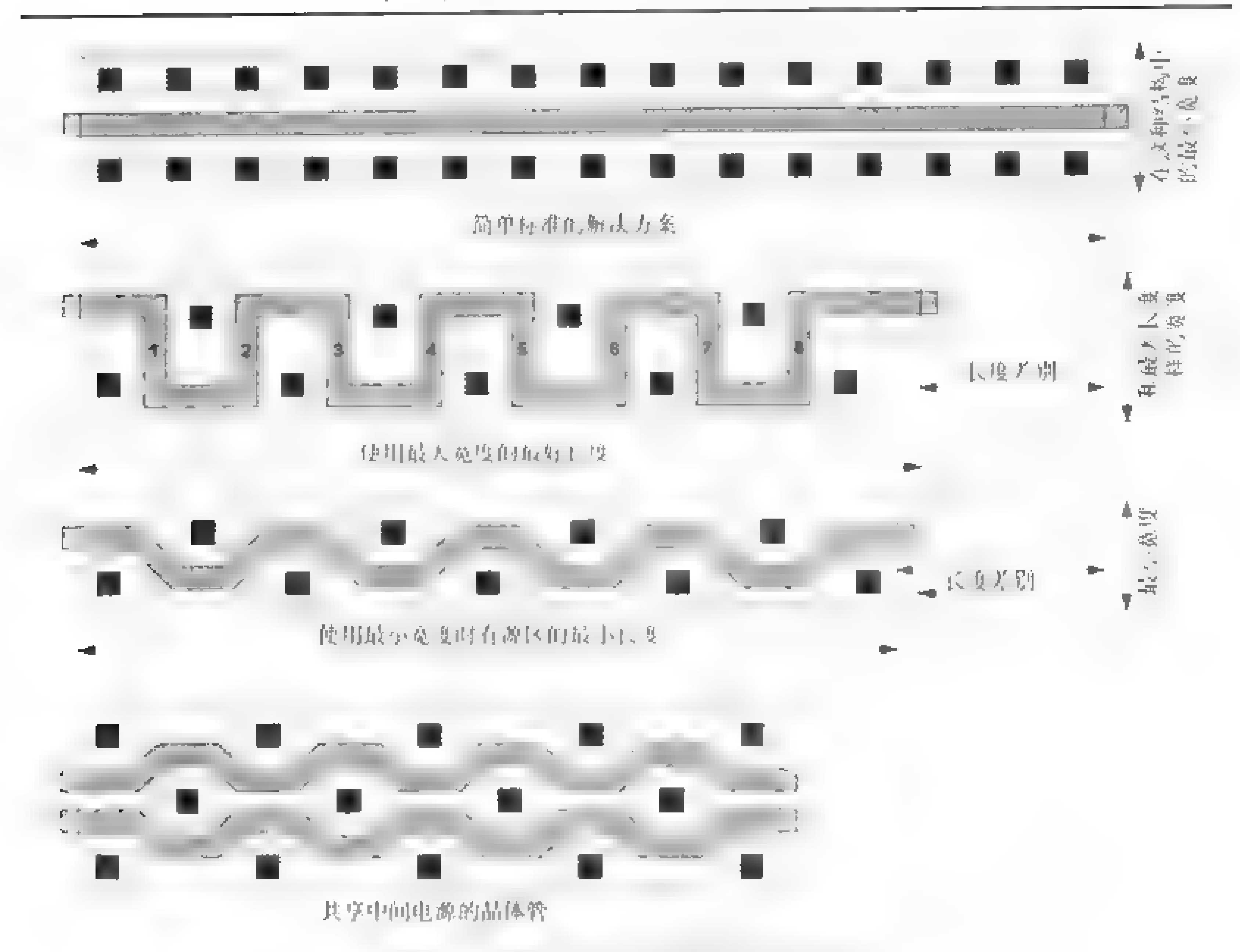


图7.26　用来最小化面积的栅弯曲晶体管

电迁移准则的应用一般要考虑三种基本情形：(a)直流（DC），(b)电子通常只从一个方向流过的交流（AC），(c)交流（AC）（双向电流）。

在理想的情形下，设计人员在设计电路时，要估计每个信号的电迁移效应。在电路图中，需要版图设计人员特别考虑的信号会被标记出来。但这种方法不是很实际，因为工作强度大而且很容易看漏和疏忽。大多数公司更愿意为版图设计人员制作参考用的简单表格，试图使运用迁移准则变得更简单。

在开始使用这样的表格之前，我们必须理解什么是单向电流、什么是双向电流，因为这和电路设计有关。这个知识点有助于我们根据电路图恰当地实现版图。

图7.27所示为一个反相器电路图和相应的版图，其中标出了电流的方向。单向电流和双向电流也都标示出了。

请花一些时间理解PMOS和NMOS晶体管中电流的方向。正如我们所能看到的，假设输出信号线为位于输出多边形右边的被切的多边形或粗体长方形。对电迁移非常敏感的区域应该是像输出缓冲器和时钟发生器一样的大驱动器。

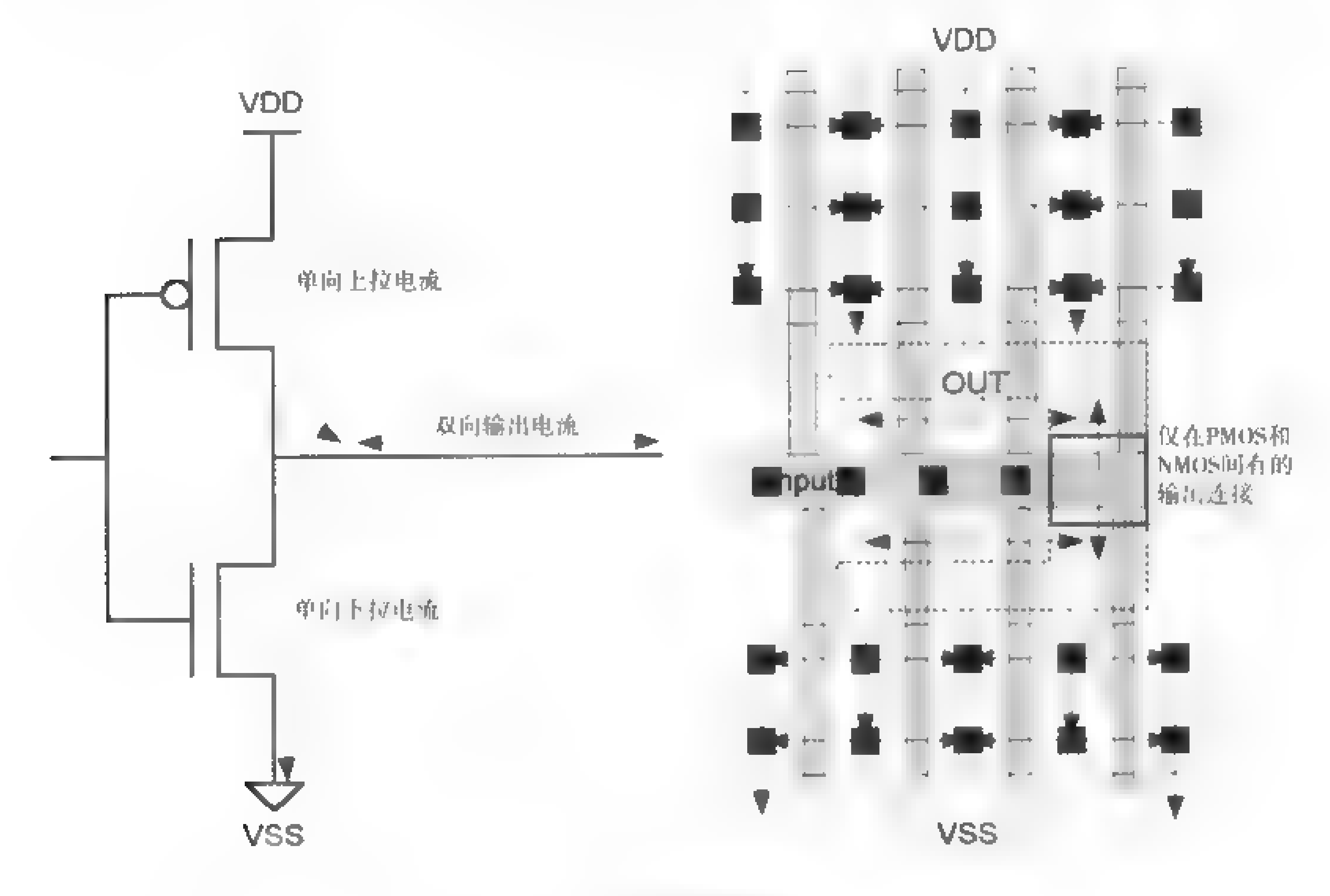


图 7.27 电迁移电流

表 7.4 所示的是一个典型的电迁移准则表

表 7.4 电迁移表的例子

驱动尺寸	平均电流	双向交流通道（Bi）（例如，信号总线）				单向交流通道（也就是 PMOS 和 NMOS 之间）			
P(μm)/N(μm)	(mA)	M1	M2	Contacts	Vias	M1	M2	Contacts	Vias
80/40	1.4	1.0 μm	1.2 μm	1	1	1.0 μm	1.2 μm	2	1
120/60	2.2	1.0 μm	1.2 μm	1	1	1.4 μm	1.2 μm	3	2
200/100	3.5	1.0 μm	1.2 μm	1	1	2.3 μm	1.8 μm	4	2
300/150	5.5	1.0 μm	1.2 μm	2	1	3.4 μm	2.7 μm	6	3
500/250	9	1.2 μm	1.2 μm	2	1	5.6 μm	4.5 μm	9	5
1000/500	18.1	2.3 μm	1.8 μm	4	2	11.3 μm	9 μm	19	9

- 如果尺寸在两个给定值之间，通常接近上限
- 金属 1 最小宽度为 1.0 μm
- 金属 2 最小宽度为 1.2 μm
- 接触和通孔的数量是基于工艺电流的通路的

注意：

- 驱动器的尺寸是有效反相器的尺寸，所以如果计算一个等价的反相器尺寸，这个尺寸也可以用于其他的门。
- 电迁移准则同样可以应用于电源节点上！这通常不成问题，因为电源线一般比较宽，连接也很结实。

7.4.3 多电源系统

现在的芯片设计师们总是力图减小他们所设计芯片的能耗，这主要是因为他们想将芯片运用在便携式仪器中。随着栅极尺寸的减小，电源电压也会降低。现在有很多标准电源电压，但是对于0.25 μm工艺，所需电源电压范围为2.5 ~ 3.3 V。

在便携式仪器的主板或封装引脚中的电源级别，不必和芯片内部工作的电源级别相同。例如，I/O驱动器用3.3 V电压，但是芯片内核工作电压为2.5 V。在这种情况下，芯片可能从PCB板上获得两种不同的电压供应，或者它有专门的内部电源管理电路，用来给外部提供3.3 V的电压，给芯片内核提供2.5 V的电压。

多电源供应的一个极端例子就是DRAM的设计。一个典型的DRAM设计需要一个已经校准的内部电压VDD，两个分离的中值电压，一个较高的电压VPP，一个I/O驱动器电压VDDQ/VSSQ，以及一个阴极衬底电压VBB。

如何有效处理如此庞大的电源系统呢？例如，如何布局布线和版图验证？下面介绍一些方法和建议：

- 按重要性和复杂性的顺序，计划好各个电源在版图中的位置，并单独进行版图规划设计。例如，有些电源已经装载进去，如嵌入式存储器中的电源，因此这种电源的优先级别就比较低。
- 记住，不同的衬底和阱进行不同的设计。例如，对于相同的P+衬底，负电压VBB和VSS不能直接连在同一个P+衬底上，所以对于连接VBB的衬底，我们必须设计其他高架衬底连线。
- 电源发生器在版图布局中很有技巧，因为在理想状态下电源应该靠近所供给的电路，这样通过减小所要求的电源网格获得理想的寄生电阻，进而提高它们的有效性。
- 在版图验证的规则文件中，要对所有电源做出准确的定义，包括器件和衬底的定义，否则，连通性检查会出现问题。
- 用分离的电源线隔离功耗大的器件 I/O缓冲器就是一个很好的例子，这样做可以隔离大量的噪声和不稳定性。
- 在相互靠近的内部电源发生器处放置存储电容。在高电流消耗期间，这些电容扮演的是电源的角色，若把这些电容放置在远离电源发生器的地方，会减小电源对电容的充电能力。

- 钳位二极管可以保护内部电源免受电源电压大幅波动的影响，如果反向连接芯片的外部电压，VDD 与 VSS 间的钳位二极管可以限制反向电压，而且，电源二极管经常是整个芯片 ESD 保护电路的一部分。

钳位二极管的另外一个特点就是当两电压差超出了一个二极管压降时，预置电压差。用这种方法，当一种电源带有大负载时，电流可以从两电源中流出。但是注意，对于 VDDQ 上的电压降，这种观点并不适用，因为 VDDQ 在带有大负载期间是与 VDD 绝缘的。

图 7.28 演示了在分立核与 PAD 电源对之间的二极管分配的情况。

电源二极管版图设计必须考虑大的前置电流，此电流在钳位情况下是存在的。

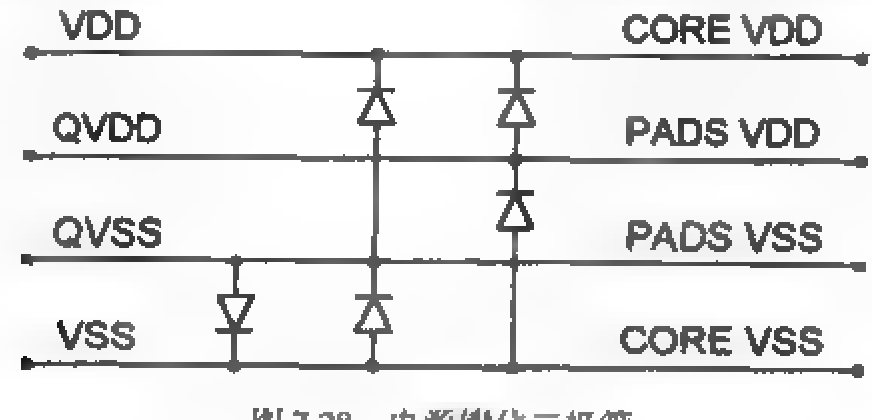


图 7.28 电源钳位二极管

第8章　考虑工艺约束的版图设计

本章将介绍一些用于解决电路中由制造工艺所引起的限制或影响的版图设计技巧。

8.1　宽金属开槽

芯片中的电源线常常设计得非常宽，以便使电迁移效应和电阻效应降低到最小。那么，金属线宽度有最大限制吗？一般来说，答案是否定的。然而，当芯片温度已经高到足够使金属显著膨胀时，非常宽的金属线会引起一个问题。

图8.1显示了金属膨胀时的热效应。当金属变热时，大块金属的侧边惯性阻止了侧边膨胀。从而导致了金属中部发生膨胀。这使得金属的中间区域向上膨起。而对于较小的信号线来说，这一效应并不明显，因为尺寸越小，侧边惯性越低，金属向上膨胀的应力越低。

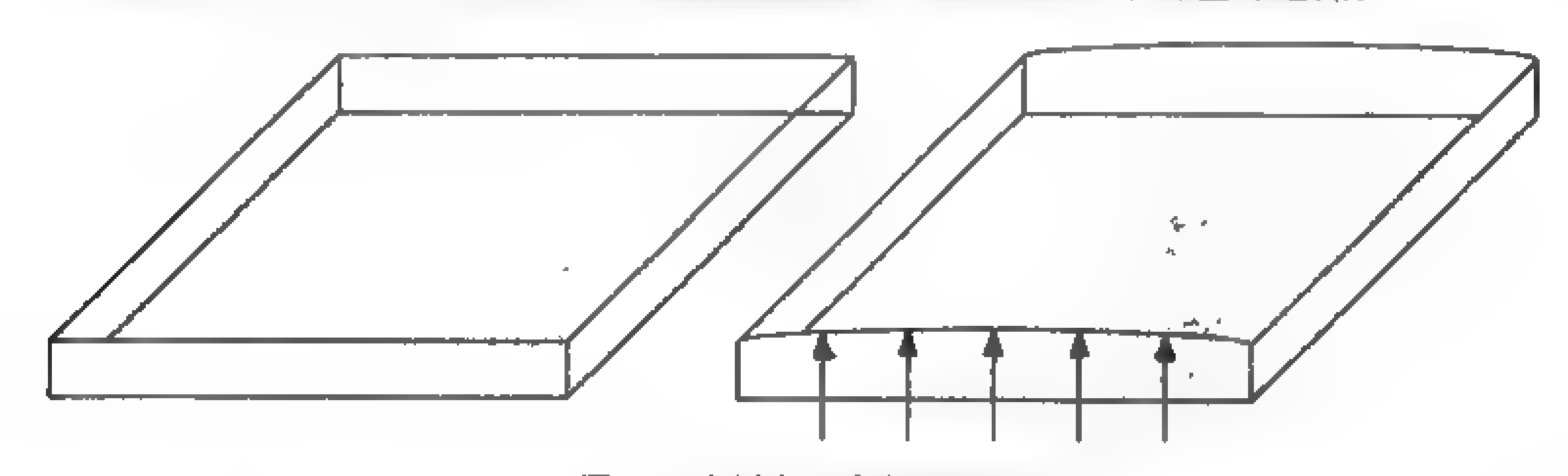


图8.1　金属由于受热而膨胀

如果金属受到足够的应力反复膨胀，金属最终会破坏保护晶圆的绝缘层和钝化层。杂质和微尘将会进入芯片，与不同的材料发生反应，从而导致芯片不能工作或工作不稳定。

为了解决这个问题，要求版图设计工程师在金属中每间隔一定距离开槽或开孔。这一方法能够有效地把一个非常宽的金属变成由许多较小的区域连接成的金属。

图8.2给出了一个带有金属开槽的金属线示例。由于开槽设计与金属的间距、膨胀温度和材料的类型有关，因此金属开槽的设计规则由工艺决定。

注意下列问题：

- 开槽的拐角处呈45°角，这可以减轻金属中大电流密度导致的压力。

- 不需要开槽的金属线的最大尺寸一般为 35 μm。
- 把特别宽的金属分成几个宽度小于所允许的最大宽度的金属。例如，如果不需要开槽的金属宽度最大是 50 μm，而在 100 μm 宽的金属线中开槽，那么恰当的做法是在金属线中插入 2 个而不是 1 个开槽
- 建立一个结构正确的单位单元并按要求对它实例化，再把开槽插入到宽金属线中的方法是最容易的。
- 开槽的放置应该总是与电流的方向一致。对于 T 型结或其他结，这一点尤为重要。
- 开槽的放置一般要超过一定的长度范围。尽量使用平均尺寸以便开槽能很容易地应用到诸如拐角和结这样的特殊区域。
- 扣除开槽的宽度后的有效金属宽度或期望的金属宽度加上开槽宽度，就是真正消耗的空间面积。例如，如果要求的金属线宽是 100 μm，插入的开槽数是 2，开槽宽度是 5 μm，则总的线宽将会消耗 110 μm 的空间。
- 在拐角、T 型结和电源 PAD 区域开槽之前要分析电流的流向。
- 如果 DRC 不能检查此处所阐述的所有情况，就需要采用目视检查

通常可以将定义开槽的设计规则很好地描述出来，但是在很多情况下使用这些规则的条件却很难被描述出来。有一些特定的区域需要用到开槽，这些区域包括芯片拐角附近的电源线和 PAD 到宽金属的 T 型连线。

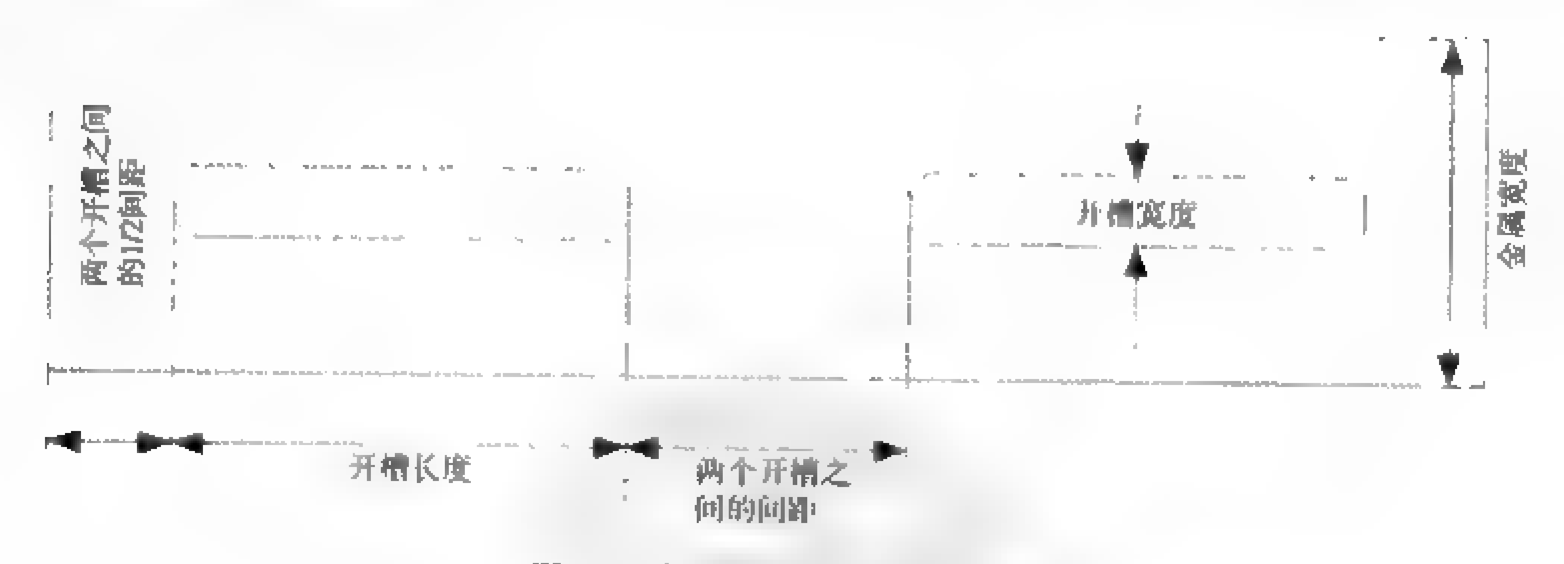


图 8.2 金属开槽实例和设计规则

图 8.3 给出了一个拐角处的电源路径，以及在这种情况下如何遵循设计规则。

如图 8.3 所示，在 45° 拐角处使用一个小的金属桥是至关重要的，因为它可以增加金属的物理阻值并防止芯片拐角在切片和封装过程中被破坏。图 8.4 给出了一个具有恰当金属开槽的从电源 PAD 到内部总线的正确连线示例。图中所示的电流可用来指导理解开槽的实现。

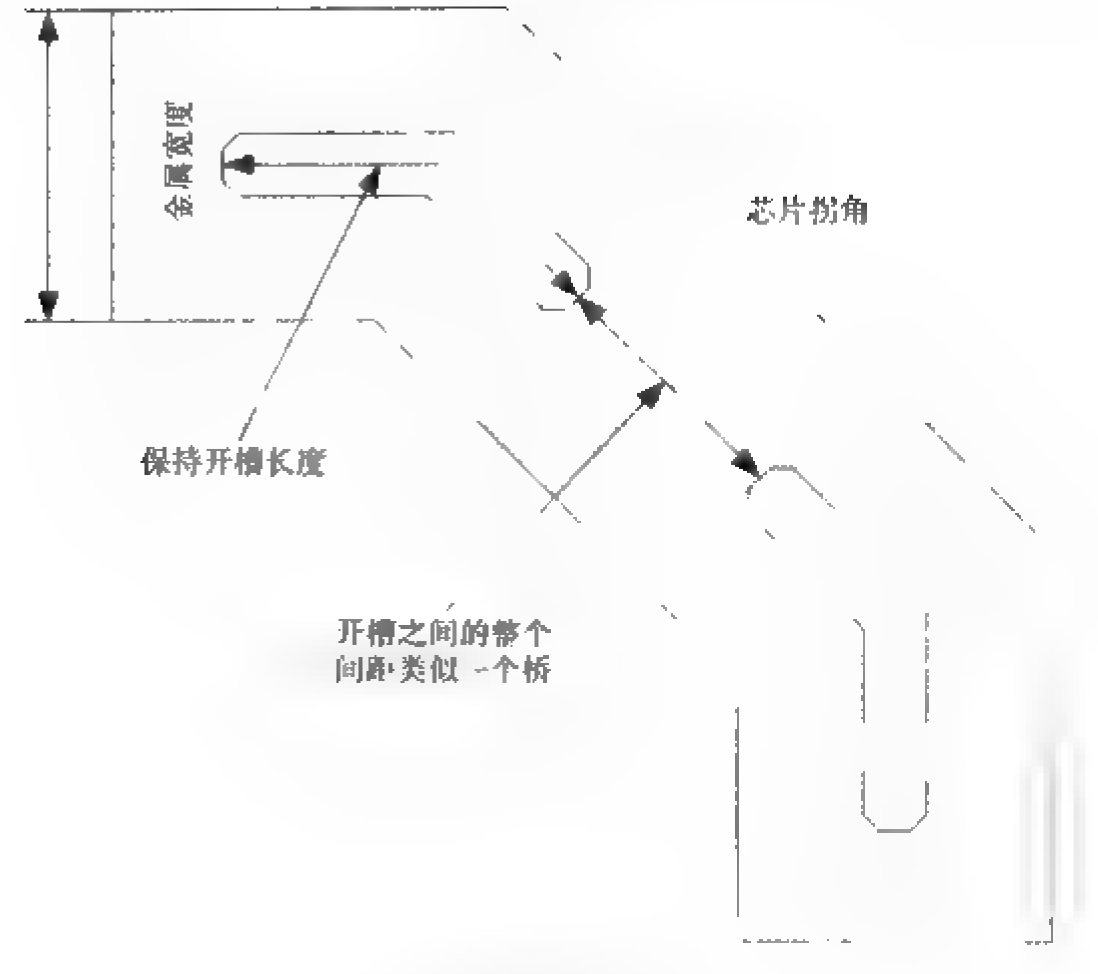


图 8.3　拐角电源布线

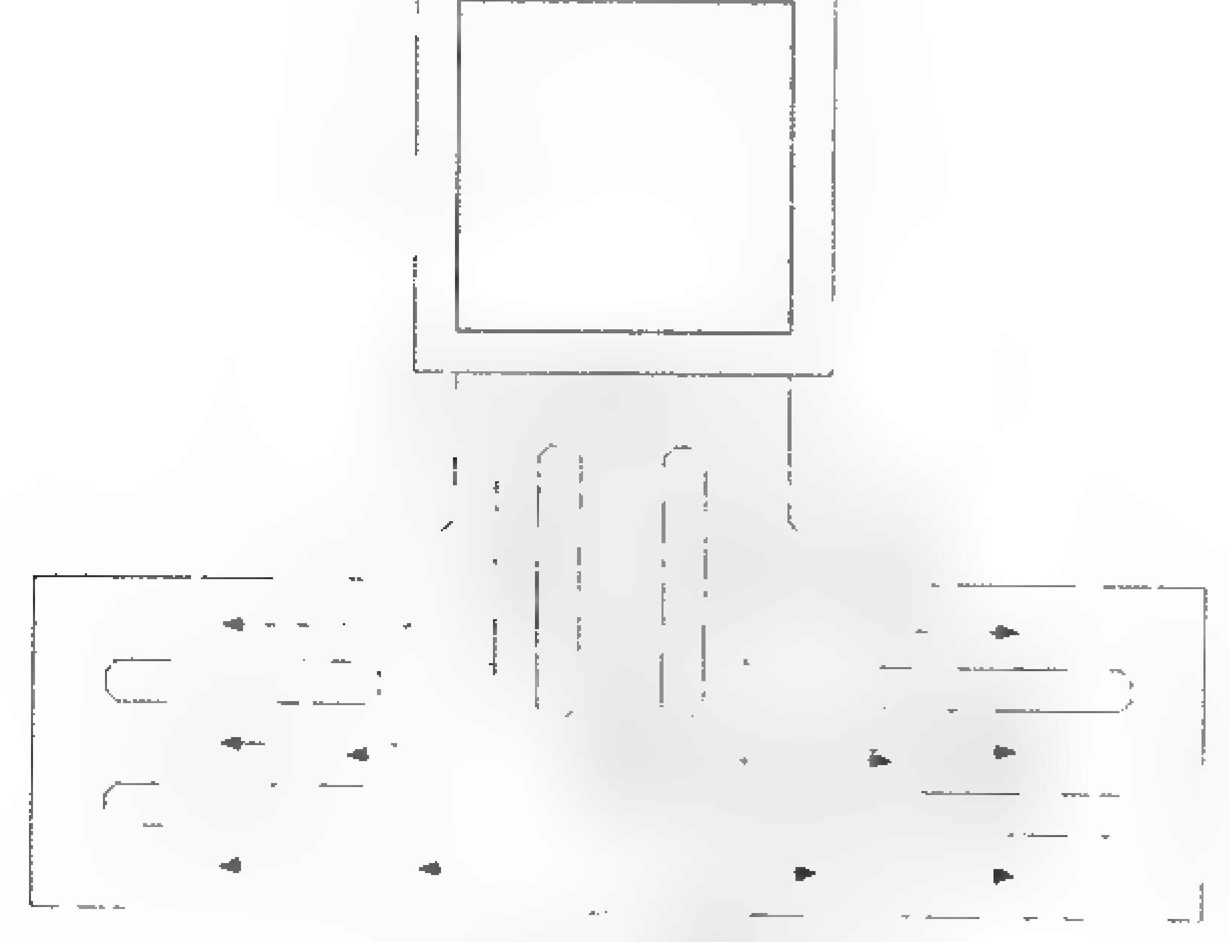


图 8.4　使用开槽的 PAD 连线

8.2 大尺寸金属通孔的实现

正如第7章所提到的，在电迁移和电阻计算中，应该考虑到将分层连接在一起的通孔。同样，还应该考虑一些与工艺相关的问题。

所谓的通孔就是指直接位于层间电流通道中的结构。因此，应该很好地掌握使用通孔作为层互连的版图设计。在大的金属线中这一点至关重要，因为一般情况下大的金属线会传输大电流。

从工艺的角度来看，通孔就是位于层间的隔离层中的开孔（hole）：要求上层的金属通过填充这个孔从而连接到下面一层。孔的制造和随后上层金属对孔的填充所形成的连接尺寸不会与版图中所绘制的连接尺寸一致。

图8.5解释了这种现象。通孔的设计规则和电学特性考虑了这些现象，目的是确保形成可靠的通孔。在本书附带的光盘中，有更多从晶圆上拍摄下来的通孔图片。

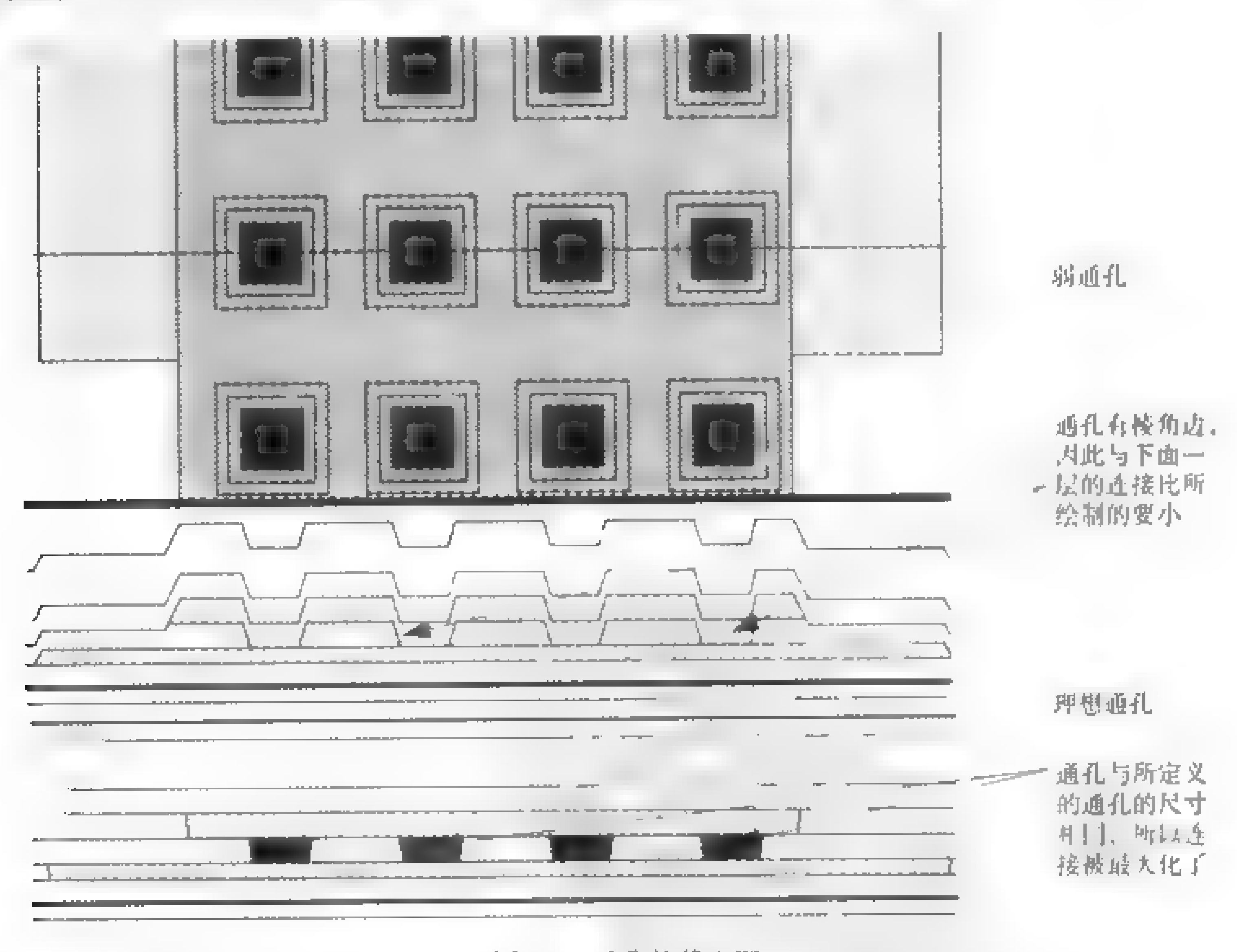


图8.5 通孔的截面图

在版图设计中，有一些技巧可以增加通孔阵列的可靠性。图8.6给出了常见的例子，而它们的有效性取决于工艺。

金属线传输电流的能力取决于金属层的宽度和厚度。这与输水管的尺寸类似。把一层与另一层连接起来的通孔数量是由多个不同的条件决定的：电迁移率、通孔电阻率、电流、工艺要求和平整度。

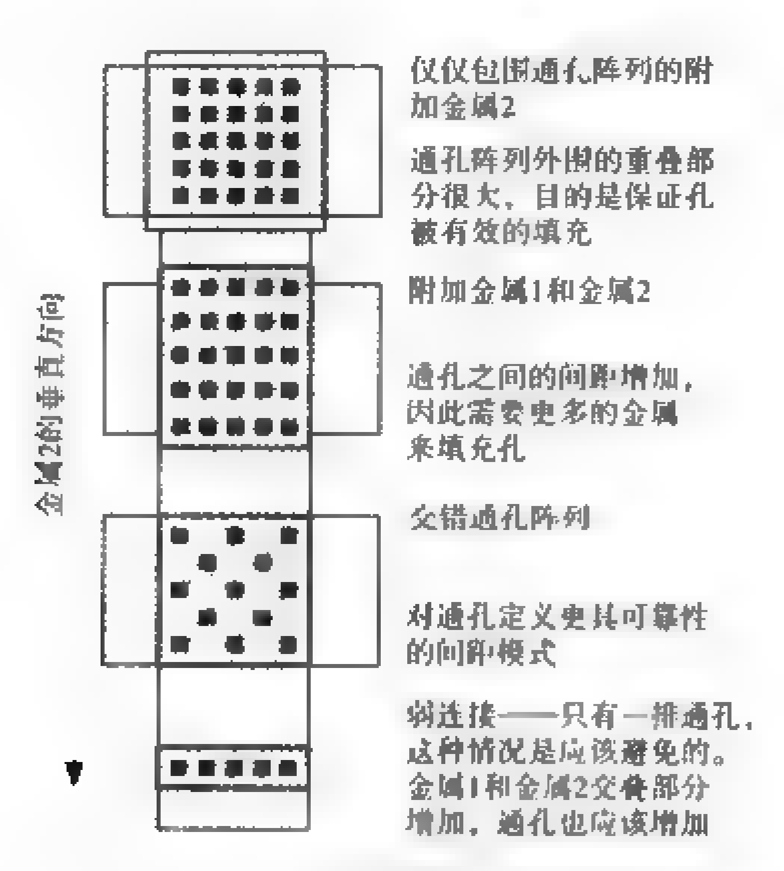


图 8.6　为工艺可靠性设计的通孔阵列结构

图 8.7 显示了针对电路性能进行优化的通孔阵列结构。

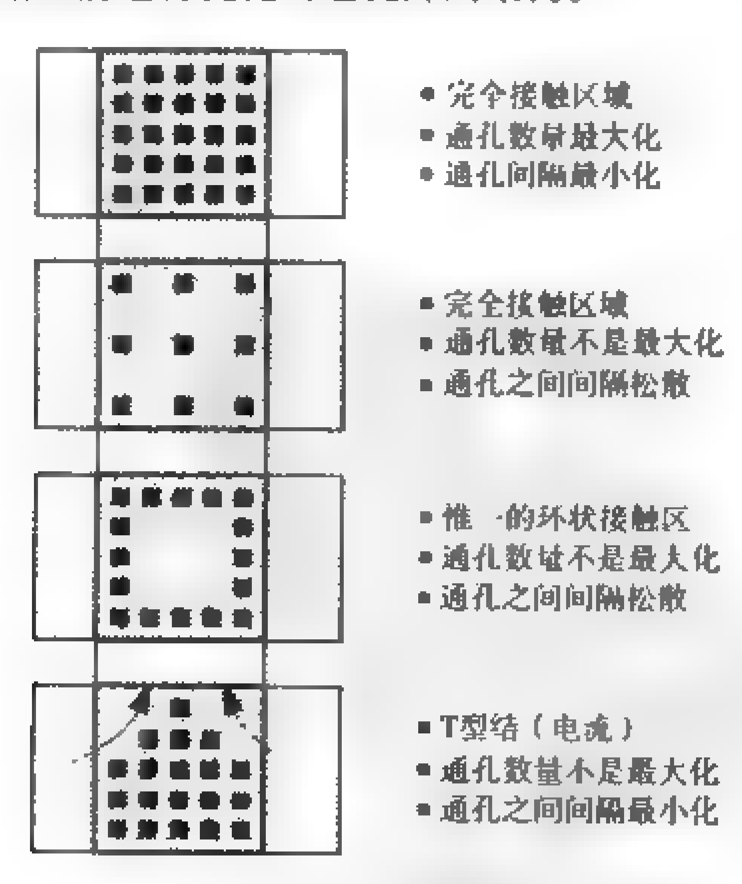


图 8.7　为电路性能设计的通孔阵列结构

8.3 台阶覆盖规则

正如我们已经阐述过的，对于不同的设计类型，例如ASIC和DRAM，所采用的工艺是不同的。基于芯片的用途、市场价格和设计要求等诸多因素的考虑，许多公司正在研究专门的工艺，而每一种工艺所对应的设计规则也在不断地发展着。

版图设计师并不涉及工艺加工过程，但他们必须采取措施，避免晶圆在化学处理和物理处理过程中可能出现的问题。台阶覆盖效应就是一种可以利用适当的版图设计技巧来解决的工艺问题。

这里所谓的“台阶”是指通过芯片特定区域层时上升或者下降的斜坡，其中特定区域是指包含有许多不同层的芯片部分。为了进一步说明这个问题，我们来看一个相对于临近区域和友好单元概念的DRAM存储器单元（见图5.23）。

和采用层间的隔离层来平整化晶圆表面一样，我们采用所谓的平整技术工艺来消除这个问题。采用这种方式，就可以消除台阶。

为什么说“台阶”是一个问题呢？如果我们分析图8.8，可以看到，当沿着多晶线对金属1进行布线的时候，由多晶高度产生的角就会生成不规则的金属截面图形。这个层也就不再是所期望的矩形，而是一个很奇特的图形。这个奇特的图形不具有和平整表面的直金属线相同的特性。在极端情况下金属线可能出现物理断路。

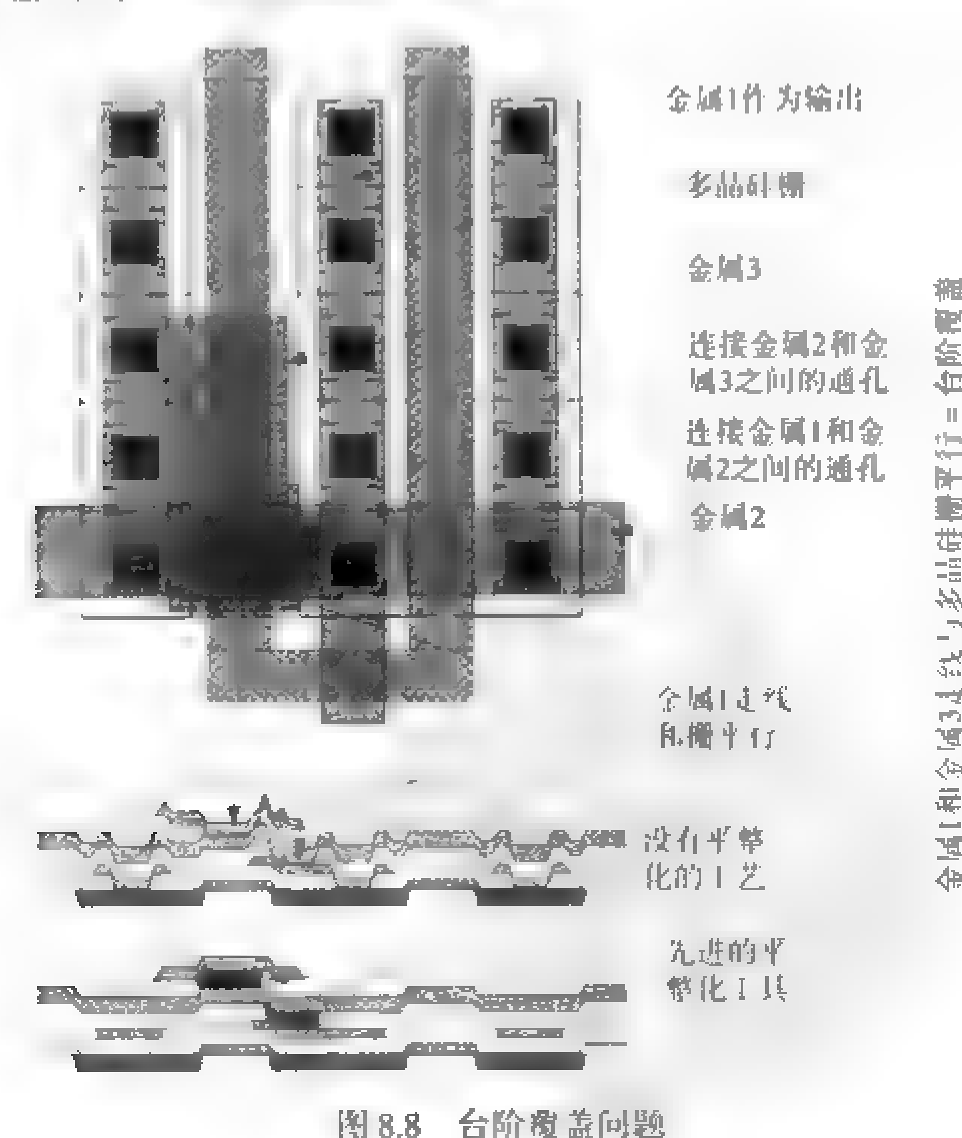


图8.8 台阶覆盖问题

举例来说，如果通孔位于不规则金属之上，这个连线就有可能不可靠或者失效。图8.8也给出了平整化的晶圆的同样情况，结果就好得多。

和平整度有关的设计规则和版图要求，就是定义特定区域上特定层的密度目标的规则，例如，对于100 × 100 μm² 面积的区域，规则指定金属多边形必须至少覆盖面积的75%。理想的情况是使用足够的多边形来确保上面层适当的平整度。

为了满足这个要求，版图可能需要使用dummy层或者附加多边形来确保整个区域中分层的连贯性。在某些情况下，设计工程师可以设定版图验证工具来自动检查密度问题，并使用dummy层来填充区域。在电学上，这些层可能是悬空的，但是一般情况下，都会被同时连接到电源线来增加它们的去耦电容。

8.4 多重规则集合

通常情况下，从事ASIC工艺工作的人设计版图时更加得心应手，因为他们有一套相对简单的设计规则。这一规则的数量比设计工程师在DRAM存储器或者嵌入式存储器中使用的设计规则要少。

为什么存储器工艺的设计规则与其他工艺相比有很大的差别呢？这是因为存储器单元占了整个芯片面积的50%～70%，而且存储器单元是依据自己的设计规则和特殊分层所进行的一种定制设计。作为一种可靠性措施，存储器在芯片上有一定的备用电路，用来替换阵列中的失效部分。

定义一套最终的设计规则是非常复杂的过程，在这个过程中，电路设计工程师、版图专家和工艺人员不得不综合考虑许多因素，如价格、尺寸、复杂性、容差、设计难度、整个流片过程工艺的可靠性等。

对设计规则而言，DRAM存储器有三种不同的规则集合：一种用于存储器单元和它的“友好”单元；另一种用于和存储器单元接口的间距相关逻辑，还有一种用于存储器阵列外部的周边版图设计；而周边版图规则和相同工艺产生的ASIC工艺规则是类似的。

表8.1给出了用于DRAM工艺的三种设计规则设置的定义的实例。

表8.1 DRAM设计规则示例

用于DRAM工艺的设计规则	单元	间距	外围
最小有源宽度	0.5	0.5	0.6
有源之间的间距	0.4	0.4	0.5
最小多晶宽度	0.25	0.3	0.35
多晶之间的最小间距	0.3	0.35	0.4
最小金属宽度	0.4	0.4	0.5
金属之间的最小间距	0.3	0.35	0.4

具有多种规则集合的版图设计必须以这样一种方式完成，即利用DRC验证工具验证不同规则集合。基本的问题在于需要对以特定设计规则进行检查的不同版图数据部分进行鉴别。

下面给出了三种样例方法：

1. 对于由特殊的设计规则集合定义的各个区域，用单独的分层设置实现。一旦版图验证完成，掩模板（mask）就由合并了的分层数据生成。虽然这种方法需要版图设计人员在不同的区域转换分层，但是一旦被完成，这些区域很容易处理。
2. 由成为模块化分层的特殊分层来定义各个区域。而所有区域的普通版图是用同一层定义的。一旦它们被完成，想要的模块化分层也就绘制好了。在版图验证工具中，生成的中间层用来区分采用不同的设计规则设定的多边形，如表8.2所示。
3. 由单元命名规则来识别区域。将版图验证工具设定为可以根据单元名来识别和区分不同的区域。在这种情况下，普通版图用一种分层集合来完成，一旦设计完成，就会为单元进行合适的命名。虽然这种方法简单，但它依赖于版图是否可以被划分为不交叠且分离的单元。

表 8.2 使用模块化层次方法生成 DRC 层

模块化层次合并方法
有源层 +MEM 模块化层 = 有源存储器
有源层 +PITCH 模块化层 = 有源中心点间距
有源层 +PERY 模块化层 = 有源 PERY
多晶层 +MEM 模块化层 = 多晶存储器
多晶层 +PITCH 模块化层 = 多晶中心点间距
多晶层 +PERY 模块化层 = 多晶 PERY

8.5 天线规则

所谓天线效应是一种会导致部件损坏的制造工艺副作用。在一定的条件下，离子刻蚀或扩散装置会使与晶体管栅极所连接的不同结构上产生感应电荷。感应电荷可能对晶体管的薄层栅氧产生过大的应力和不可恢复的损害，从而导致晶体管工作不稳定。

如果结构被建造的像天线那样运作，那么在制造过程中，电荷就很容易被感应。图8.9显示了一个天线式结构的实例，在这个结构中，场区（厚氧）上的多晶和栅（薄氧）上的多晶的比例足够大，对每种工艺还有其他的特殊规定，包括对金属的规定。

随着栅尺寸变得更小、芯片中增加的金属越多，以及工艺工程师减少氧化层的厚度，天线效应对晶圆的成品率会有更大的影响。

实例中所给出的方法1描述了一种消除天线效应的方法，即通过将有天线作用的长多晶分解来消除天线效应。

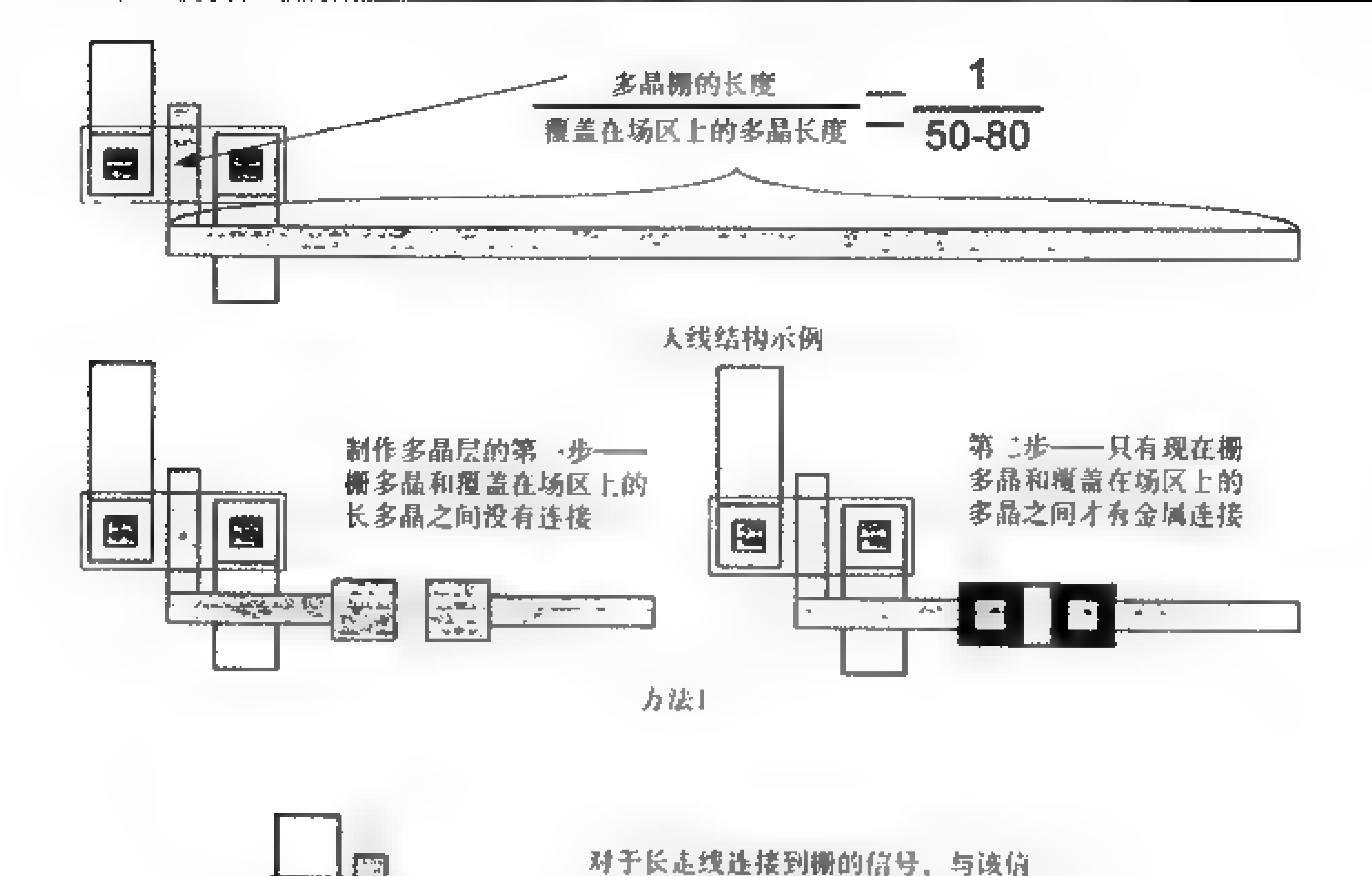


图8.9　天线规则示例与解决方法

而方法2给出的方法是通过在有可能引起天线效应的晶体管附近放置二极管来消除金属天线的影响。一旦由于天线效应而在金属上感生出足够多的电荷，二极管就会将这些电荷转移到衬底上。

在许多先进的工艺中，由一定数量的布线层（如金属1、金属2等）组成的结构需要进行天线规则检查。一些自动布线工具知道如何限制金属的长距离走线，从而避免天线问题。

如果内部固有的二极管连接到通道连线上，就不需要在特定的节点添加外部、独立的二极管。例如，另一个晶体管源或漏区域完全可以像二极管一样，转移走不必要的电荷。

8.6　特殊设计规则

前面介绍了有关工艺顺序流程、与工艺、电路要求相关的设计规则等基本知识。这里还将介绍一些值得注意的不同于“一般”设计规则的特例。

尽管本书不能详细地解释所有与特殊工艺类型相关的“奇特”规则，但读者依然可以通过本节了解一些与工艺相关的、在有关设计规则的开始部分没有涉及到的问题。通常，与从事工

艺的人交流对于理解采用特殊设计规则的技术原因是很有意义的，而且与他们一起工作又可以解决实际的问题。

8.6.1 最小面积规则

随着几何尺寸越来越小，"最小面积"层规则成为了更为流行的规则。虽然设计规则的尺寸不断减小，如晶体管的栅长，但在制造过程中，并不是所有的层都能同样缩小。

对小的多边形而言，有源区面积的定义就是一个这样的示例。这种约束是作为最小宽度和面积定义的特定规则。在金属1和有源之间由单个接触孔实现的衬底连接中，小的有源矩形出现的频率相当高。在这种情况下，如果对多边形长度和宽度，仅需要遵循最小宽度规则，有源多边形将会比所生产的更小。

增加多边形长度是一个解决符合"最小面积"规则的简单方法，图8.10所示的版图阐述了这个规则。

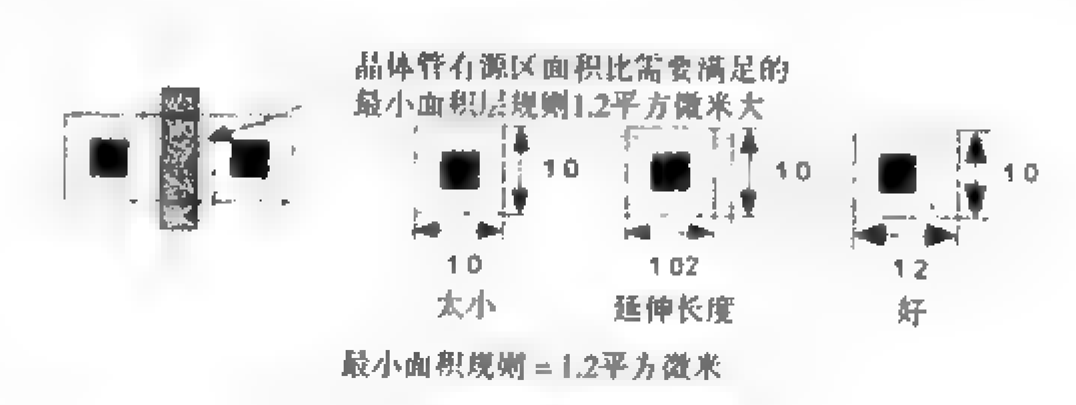


图8.10 最小面积层示例

8.6.2 末端交叠规则

虽然第3章介绍了一种接触孔覆盖的通用规则，但是一些工艺中用"末端交叠"规则进一步加强这个规则。

该规则应用于接触孔位于线最末端的情况下。图8.11中虚线标出的"硅上图形"说明了这种规则所要解决的问题。在实际的硅晶体上，当金属被包围时，位于线末端的接触孔存在没有被填满的危险。如果在制造过程中出现任何没有对准的情况，这种危险就会增加。

8.6.3 双接触孔

再次考虑DRAM工艺，其中制造工艺的约束一直被测试来获得最小的存储器单元。在这种情况下，对于一些设计规则来讲，许多标准化结构的整体可靠性都存在一些疑问。

在存储阵列外，版图的拓扑图形不再是规则的，并执行更严格的设计规则。其中之一就是每种连接都采用双接触孔或双通孔要求，此设计规则应用于晶体管版图、一般的布线以及信号连线。

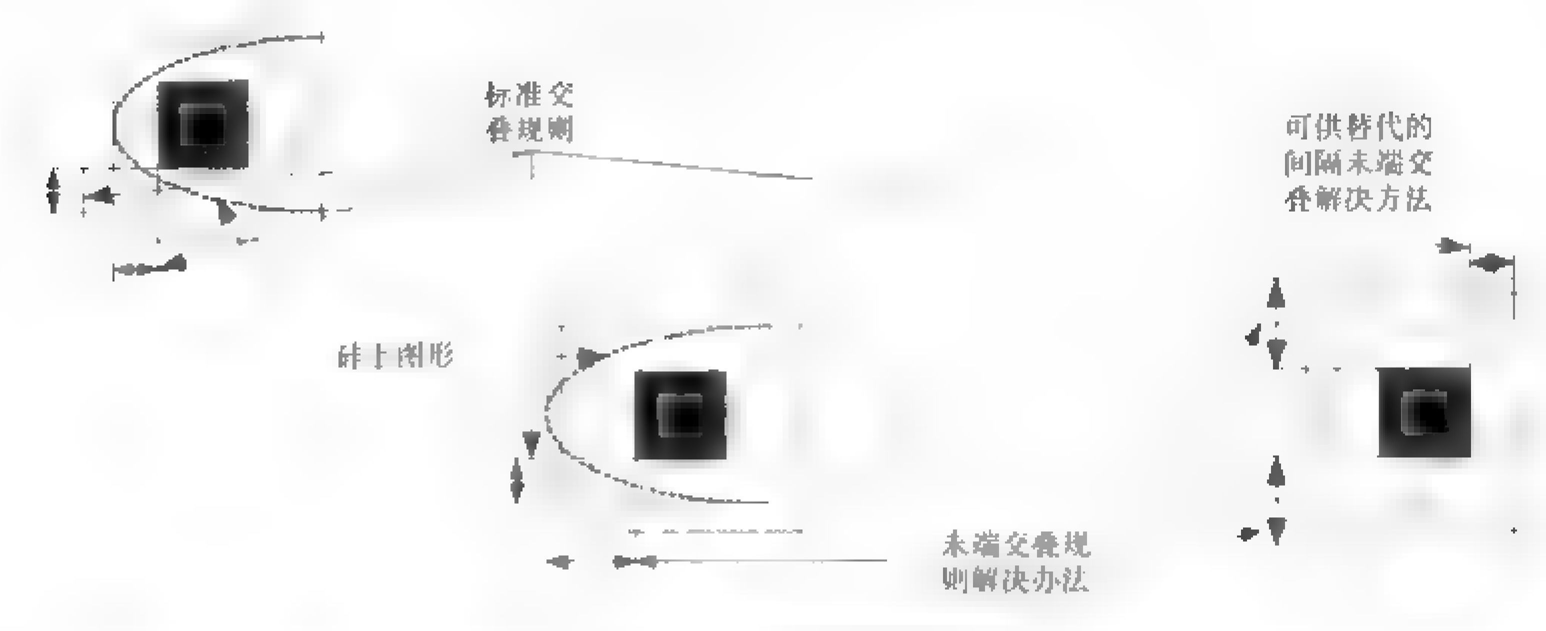


图8.11 末端交叠规则示例

双接触孔和/或双通孔不仅仅改进了连线的电阻；更重要的是，为每条连线提供了额外的接触孔，从而提高了可靠性。

这种要求产生的问题是，用自动工具来放置双接触孔对布局布线工具是一个挑战，这些工具更喜欢方形的通孔/接触孔单元。X轴和Y轴方向上的连接是很容易实现的，而且在布局的过程中，工具不需要确定接触孔和通孔单元的方向。

图8.12给出了两个等价的版图设计，一种仅采用一个切口单元，另一种采有两个切口单元。从这个示例可以看出，这种版图风格增加了芯片的尺寸和设计存储器芯片需要的工作量。目前，DRAM版图设计工程师很难利用自动化工具。



图8.12 单个和两个接触孔的设计方式

8.7 闩锁效应

就概念来讲，闩锁效应指的是由于寄生效应引起的VDD和VSS短路，从而产生的集成电路的一种不可恢复的状态。根据闩锁状态的严重程度，集成电路可能被永久性的损坏，或者只有在电源彻底切断后才有可能恢复。

下面简单介绍一下闩锁效应是如何产生的。图8.13给出了简单的CMOS反相器中寄生器件的等价电路模型。传输门是闩锁效应的另一个危险源，特别是当传输门的源或者漏与VDD或者VSS连接时。

重新绘制图8.13所示的电路图使其更为易懂，而得到了图8.14。很简单，只要两个双极晶体管中的任一个导通，闩锁就会发生。如果这种情况发生，由一个开启晶体管产生的电流会使另一个也开启，因而电路中就会产生正反馈环。同时第二个晶体管的电流就会加强第一个晶体管的驱动能力，并且使自身产生恶性反馈，从而导致正反馈再次发生。在正常情况下，闩锁效应是不可能发生的，那么这些晶体管是如何被激发的呢？首先，简单地讨论一下双极晶体管是怎样工作的。

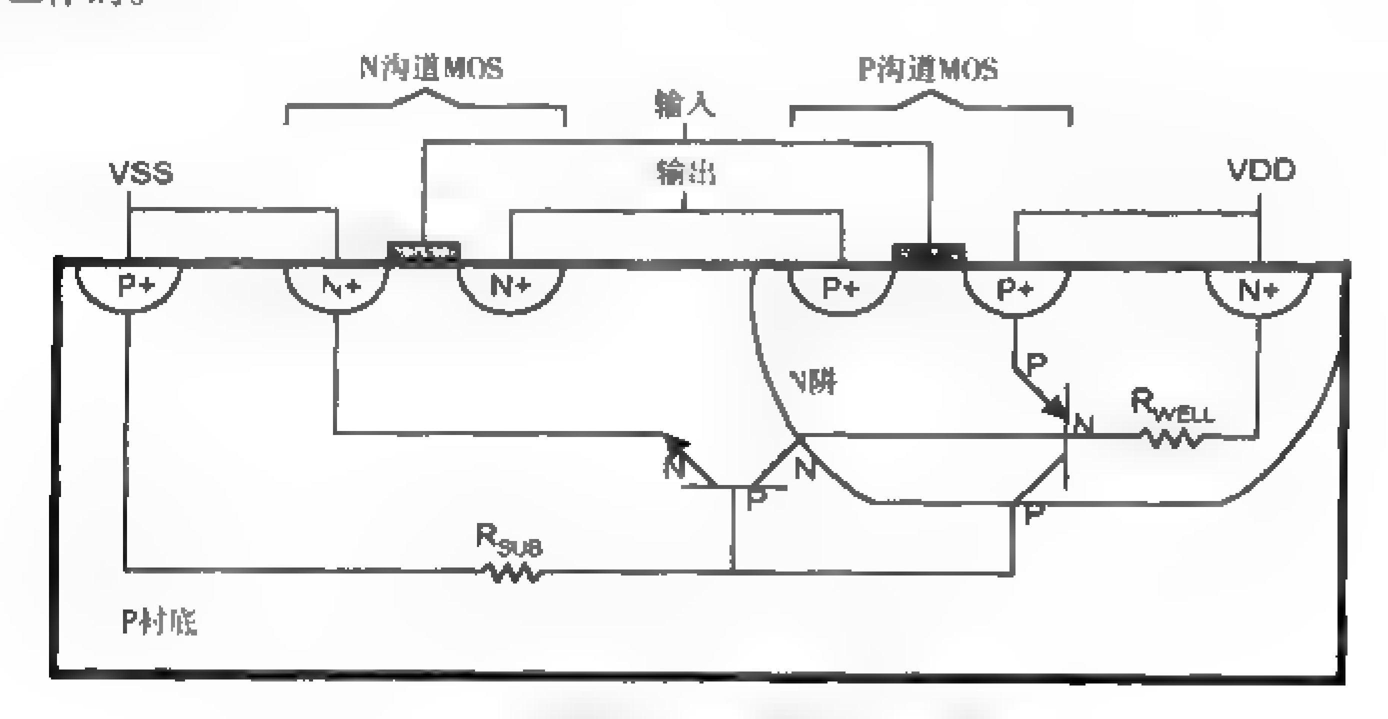


图8.13 有闩锁效应电路模型的反相器剖面图

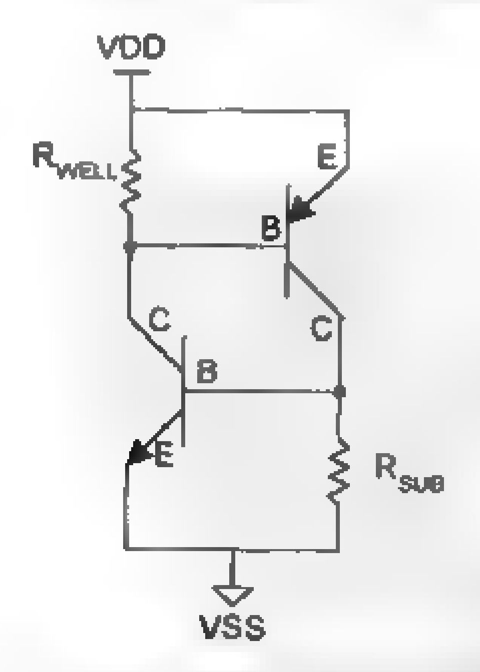


图8.14 重新绘制的闩锁效应电路模型

同MOS晶体管一样，当基极（标记为B）和发射极（标记为E）之间有一个电势差的时候，双极晶体管就会被激发。NPN双极晶体管需要正的VBE，而PNP晶体管需要负的VBE。在激发NMOS和PMOS晶体管时，会看到晶体管和电压要求的相似功能。

在正常情况下，如果寄生双极晶体管是关断的，那么是什么触发闩锁效应呢？触发闩锁效应的最普遍情况是不想要的电流或极端的电流通过电源电压VDD或VSS注入到芯片。这些节点直接连接到外部，从而总是会有不确定的电压和电流。

重新分析图8.14的电路图，假设有外部的异常电流注入到VDD。根据欧姆定律，该电流会导致RWELL上产生电压降，从而晶体管获得的负VBE使PNP晶体管导通。PNP晶体管生成的电流会在RSUB上产生另一个电压降，导致NPN晶体管开启。即使外部的VDD电流消失，NPN晶体管电流也可以维持RWELL上的电压降，保持PNP晶体管的开启状态。反之，PNP电流维持NPN晶体管的工作。这个恶性循环就是闩锁效应。

综上所述，以下就是已经讨论过的闩锁效应发生的基本条件：

- 产生足够大的电压V_{BE}（至少是瞬时的）来激发任意一个双极型晶体管。这需要注入芯片的异常电流和足够大的寄生电阻，从而根据欧姆定律能够产生这一电势差。
- 寄生双极晶体管有足够电流驱动能力来维持所需的V_{BE}，从而保持双极晶体管开启。

闩锁效应是一种已经被完全掌握了的现象，因为它在CMOS集成电路设计中已有很长的历史了。现在已经提出的许多指导方法和设计规则都可以从本质上减小闩锁效应发生的几率，并减小所列条件发生的可能性。

一种降低闩锁效应可能性的直接方法就是减小穿越基极和发射极节点的寄生电阻，从而提高能够触发闩锁效应所需的电流强度要求。如果这个方法实现恰当，那么能够触发闩锁效应的电流就可能大到在物理上很难实现。

一般情况下，这些技术通过把芯片分成许多小块来减弱双极晶体管的驱动能力，因此缩小各种双极晶体管的有效尺寸。

减小闩锁效应的版图方法包括以下方面：

- 避免在电阻性材料中，比如扩散区或者多晶硅中，进行电源线布线（特别是连到衬底或者TUB接触孔的电源线）。保持电源节点在金属中！
- 把衬底和TUB接触孔放置在不同类型的晶体管之间，此外，要使衬底接触孔和带阱的晶体管之间的距离最小化，反之亦然。例如，如果PMOS晶体管在一个N阱中，那么就把P型衬底接触孔和PMOS晶体管放得尽可能近，在N型TUB接触孔和NMOS晶体管之间采用同样的逻辑。

- 使衬底和TUB接触孔的数量最大化。
- 使衬底和TUB接触孔之间的距离最小化。
- 保证整个芯片上的衬底和TUB接触孔覆盖均匀。
- 使用连续的条状或者带状衬底和TUB接触孔。这种技术在形式上称为保护环，尤其是指这一带状接触孔完全围绕在晶体管区域周围的时候。
- 把同种类型晶体管分放在一起，以避免由于需要防范很多不同区域的闩锁效应而带来的工作消耗。
- 内部电路尽可能远离外部PAD区域放置。
- 在衬底或阱区与晶体管源端的电势不同的区域，尤其要注意闩锁效应。

在所有情况下，最好可以形成由版图验证工具检测的、正式的量化设计规则。虽然上面所列的许多方法并不是很容易转换为数字化的检测，但是建议至少应有如下方面：设计规则应该明确说明衬底或TUB接触孔之间的最大距离。

通常，这个距离大约是40 μm或50 μm，但是随着晶体管栅长的缩短，此准则也应该减小。这个准则要确保给定区域中基底连接的均匀覆盖，同时整个区域上有一致的电压。

这里有检测这一距离的两种基本方法：从有源到有源或者从接触孔到接触孔，如图8.15所示。根据测量它们之间距离的数据可见，这两种方式都是正确的。

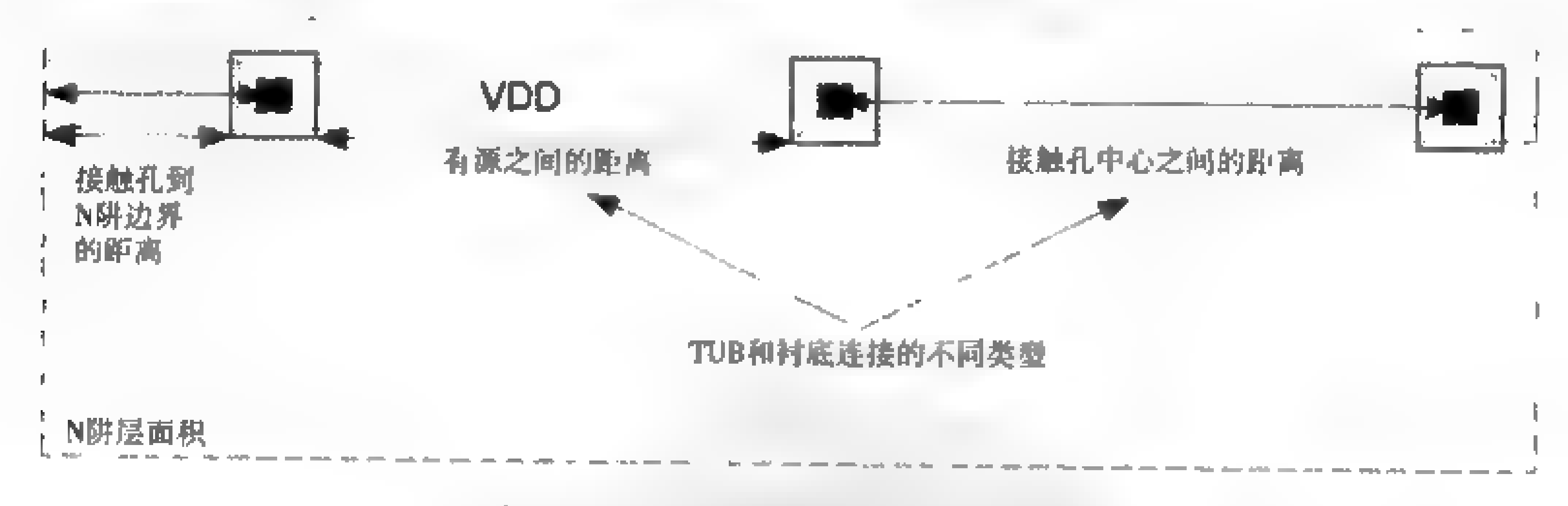


图8.15 TUB连接到N阱和VDD电压

第二条规则是基于衬底或者N阱电阻决定每种接触孔的面积覆盖。每一个有源区都应该有衬底覆盖它们之间最小距离的一半以上来防止闩锁效应。所以，如果最小距离是50 μm，覆盖应该是25 μm。

从图8.16可以看到，当版图与给定的规则一致的时候，在四个接触孔之间的距离中，会留下一个晶体管区域内不可保护的小岛。用黑线标出的每个接触孔的阴影区域就是这四个TUB接触孔的覆盖范围，中间白色区域则是没有被覆盖的部分。

经过完全验证的版图必须符合这两个规则。

大的晶体管或可以快速连续转换的高速电路根据尺寸会注入大量的噪声到衬底中,时钟发生器或者输出缓冲器就是这样的例子。在晶体管周围的所有TUB和衬底连接上使用完整的保护环是一个明智之举。

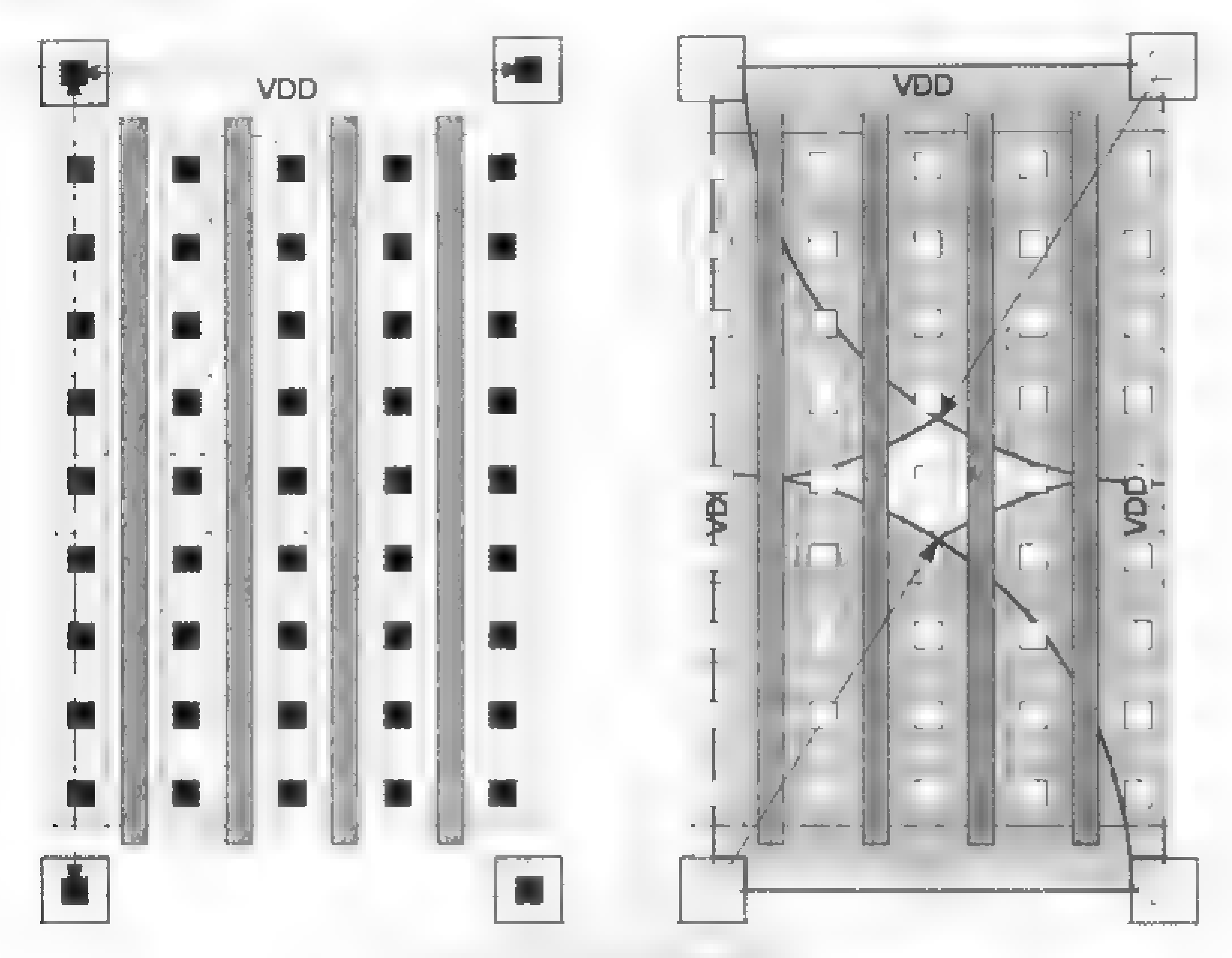


图8.16　N阱内TUB覆盖区域

版图中保护环被分成两种基本的类型:硬环和软环。硬环指的是一个完全接触的有源环,在这个环上有连续的金属1。而软环只有一个连续的有源环,但是金属层可能被打断以使信号通过保护环。

图8.17阐述了硬环和软环之间的不同。

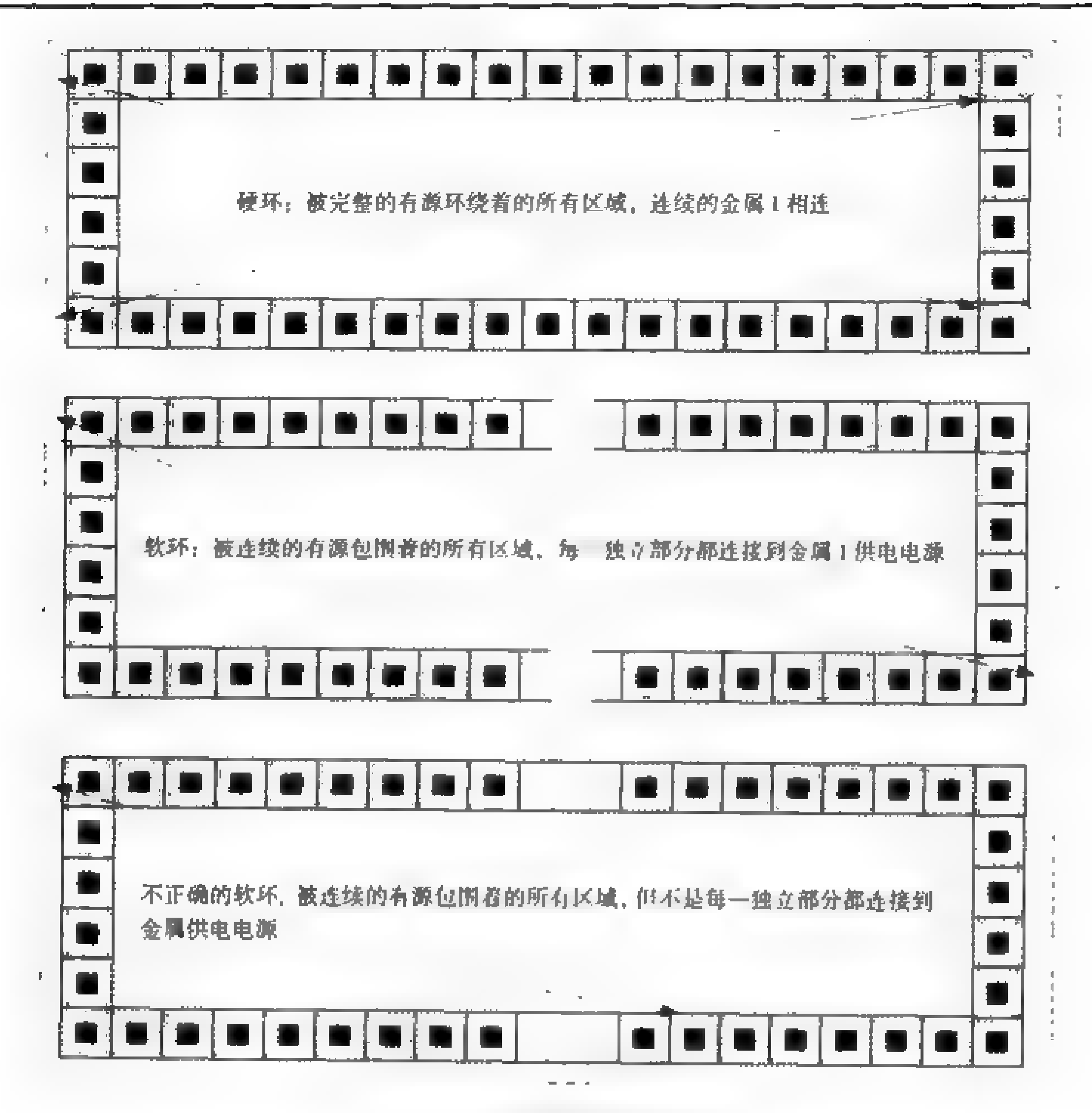


图8.17　硬环和软环示例

第9章　不确定环境下的版图设计技术

我们可以使用很多版图技术对电路设计进行期望的修改。对版图进行最终的修改是很正常的，因为对任何工程项目来说，都不可能有充足的时间在提交版图前对它进行充分地验证。考虑到必不可少的修改过程，本章将讨论的版图技术，可以最大程度地减小修改对整个芯片版图的影响。

9.1　便于修改的电路版图设计

虽然不能预先知道哪些电路需要修改，但是在许多情况下，我们能够预测有些电路是需要进行调整的。为了方便调整，可以从开始电路设计时，在电路图中放置一些用来调整电路功能或者时序的特定结构。这里将推荐的版图技术，可以用来实现这些电路设计。

9.1.1　可编程的金属层配置

可编程的金属层配置是一种与门阵列设计中所包含的理念类似的方法，但是它应用于特定的场合。其设计思想是，在设计中有一套通用的基本电路和版图，并且可以根据不同的金属掩模层配置来编程。

版图设计时应考虑到便于今后的修改，可编程金属层配置就是这一概念在单个电路上的实际应用。母片是门阵列设计中的一个概念，它指的是整个芯片都是可编程的。金属层配置指的是可以小规模地修改或定制一些具体的设计，比如延时电路、控制逻辑或者存储器的配置。

金属层配置的应用实际上是非常广泛的，而限制它使用的反而是设计者的想像力。下面给出一些使用金属层配置的例子，是很有价值的：

- 重新配置设计的功能。此时，设计中包含一套通用基本分层，不同的金属层配置会产生不同的终端产品，下面列举一些可以被重新配置的特性：
  操作特性，如低功耗模式和测试模式
  I/O 接口标准：TTL, LVTTL, SSTL
  I/O 数据宽度：×1，×4，×8，×16，×9，×18
  电源电压：1.8 V, 2.5 V, 3.3 V, 5 V
- 模拟电路微调，比如阻值和器件尺寸。在第6章中已经讨论过 ESD 电路调节的例子。

- 对照确定的电路设计来评估新的或未经证明的电路。比如，在一块芯片上实现两款设计，通过金属层配置实现二者之间的切换，从而比较二者的差异。这对于需要技巧的模拟电路，如锁相环（PLL）、输入缓冲器，以及振荡器等是非常有用的。
- 通过接通或断开器件、单元或逻辑门来实现电路的调整。

下面详细探讨一个非常普遍的例子——延时链。在延时链的例子中，金属层配置常用来解决很多不同的、灵敏电路的设计问题，因此非常有用。

图9.1给出了同一个延时链电路图的不同设计方案，它可以在0到3个延时级之间进行配置。

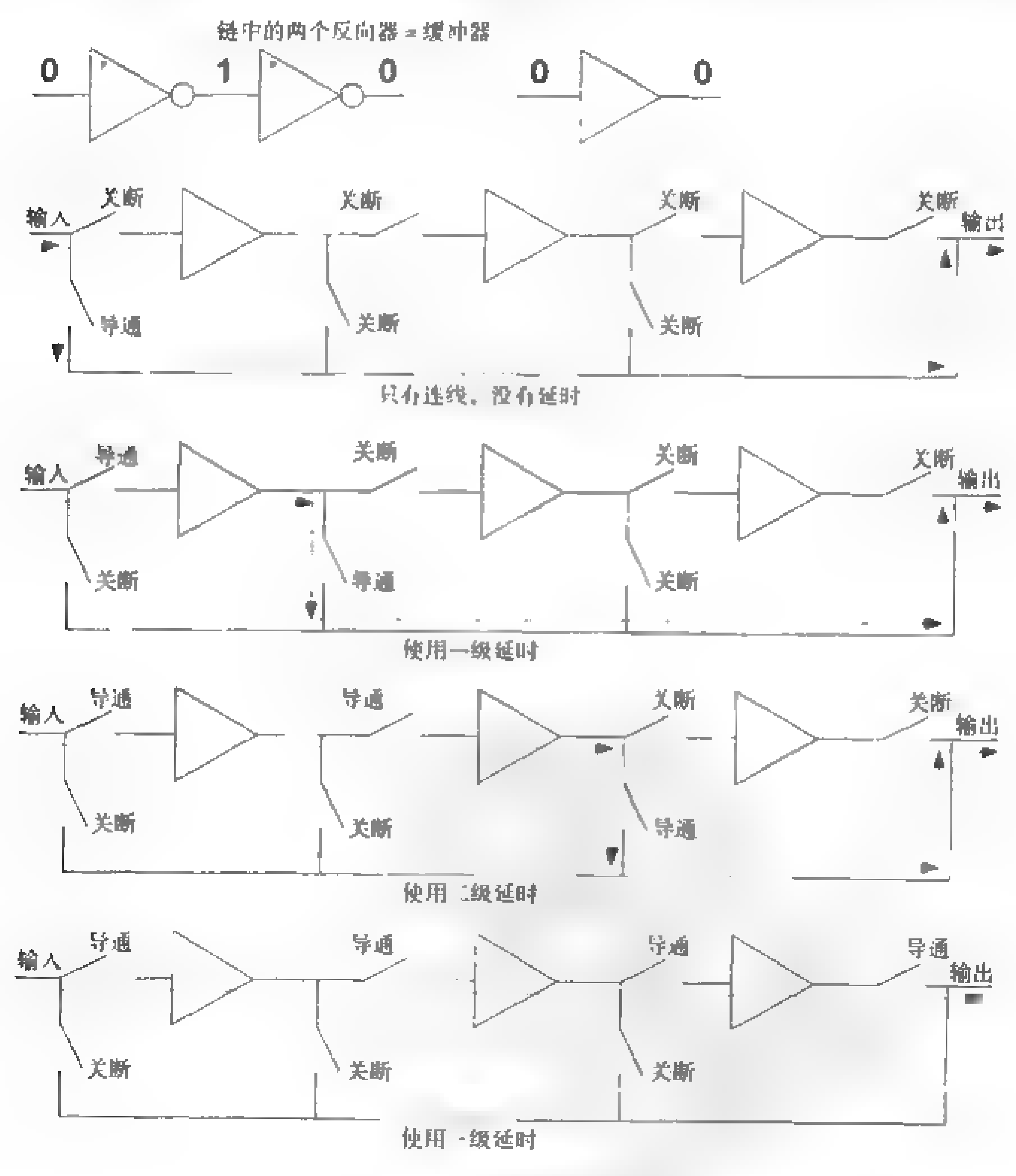


图9.1 可配置的延时链示例

电路图中的导通或关断开关和电路的短路或开路相对应，虚线表示各种情况下的延时路径。在设计流程中怎样实现和使用金属层配置呢？

1. 电路设计工程师提供已经绘制好的电路图，在该电路图上正确绘出金属层配置。
2. 定义两个与电路中的OPEN开关和CLOSED开关相对应的附加层，这些层必须和实现这些配置的掩模层相关联。

例如，对于两层金属工艺，通常在顶层金属层上实现配置，因此，配置应该和金属2相关联。

使用单个的分层和器件时，允许使用布局布线验证工具，一方面验证连接关系，另一方面确保配置遵循一些既定准则，而这些准则应该能够反映DRC要求和产品测试要求，在这里配置区域可以再次使用。

无论是绘制配置层还是对其进行DRC规则检查，都必须小心谨慎，因为CLOSED层可以在任何时候改变为OPEN层，所以版图设计和检查必须适应这种灵活性。

在这些层上绘制的多边形被认为是和电阻类似的器件，必须如电路图所示进行连接。图9.2给出了一个版图示例。

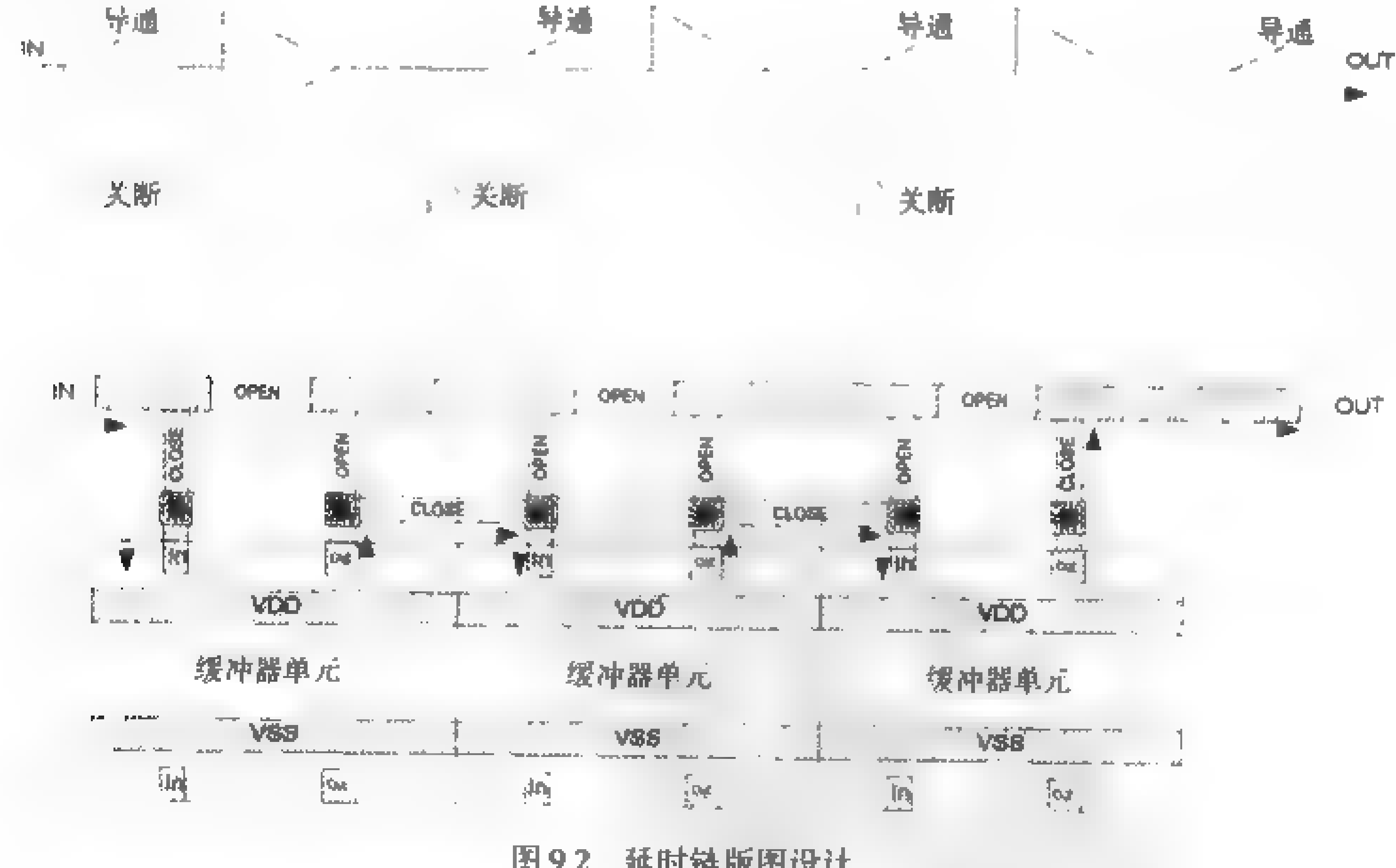


图9.2 延时链版图设计

采用金属层配置的设计可以在最终配置完成前，就先开始进行版图设计。一旦版图设计完成，就可以简单地从一种配置转换到另一种配置，因为金属层配置的改变并不影响单元的面积和拓扑结构。

1. 当为交付或制版生成最后的GDSII文件时，CLOSED层就会和相关的掩模层合并到一起（在此例中为金属2），并舍弃OPEN层。
2. 在原型测试期间，产品测试工程师可以修改电路的硅实现。一种方法是用聚焦离子束（FIB）机再次改变电路。该机器可以使CLOSED配置点关断，或使OPEN配置点导通；另一种方法是用激光切断CLOSED配置点。
   前面提到的配置层的准则必须包括一些限制，而这些限制是由使用在硅原型的方法带来的。
3. 一旦利用流片验证结果来确定最终配置，就必须更新电路图和版图数据库来反映最后的设置。

应注意的是，仅对那些可以进行FIB修改的特定芯片才能进行FIB修改，因此，如果需要修改掩模层，必须更新版图数据库并重新生成所需的掩模层。

应该根据更新的数据库来制造产品的掩模层。

9.1.2 通孔的可编程性

谈到便于修改的电路版图设计，通孔编程应用于真正设计成可编程的电路中，比如只读存储器（ROM）或者专用解码器，它们可以基于基本设计进行独立的配置。

在这种情况下，电路配置由“编码”方案决定，只需在可编程电路的顶层放置通孔才可以实现这些通常由软件开发组设计的编码方案。

为了实现版图设计的通孔可编程性，需要考虑两个方面的问题：

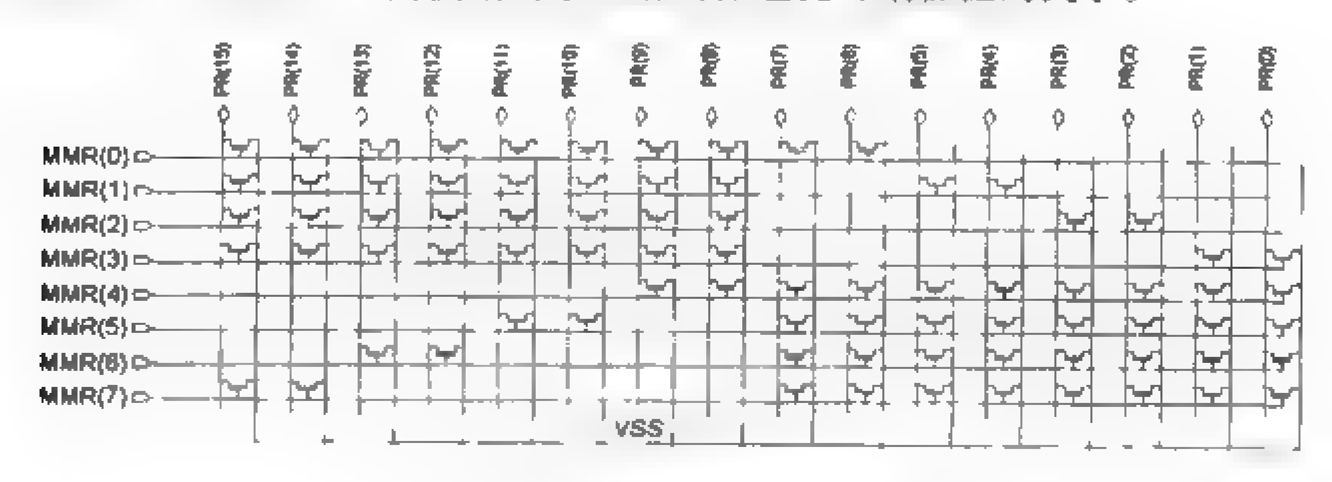
1. **未配置的版图设计，要能够适应各种不同的通孔放置方式。**“合法的”位置应该设计在子单元和组合的设计中，用于通孔布局的合法位置应该使得最终设计在任何情况下都满足设计规则的约束。
2. **把通孔的布局或编程纳入到基本版图设计中去的方法。**通常是一个采用多边形编辑工具作为版图设计引擎的宏单元，或者一个定制的CAD工具包，它可以读写标准格式的版图数据库。

在如今的存储器设计中，用软件进行通孔可编程性设计是一种很通用的技术，它具有以下优点：

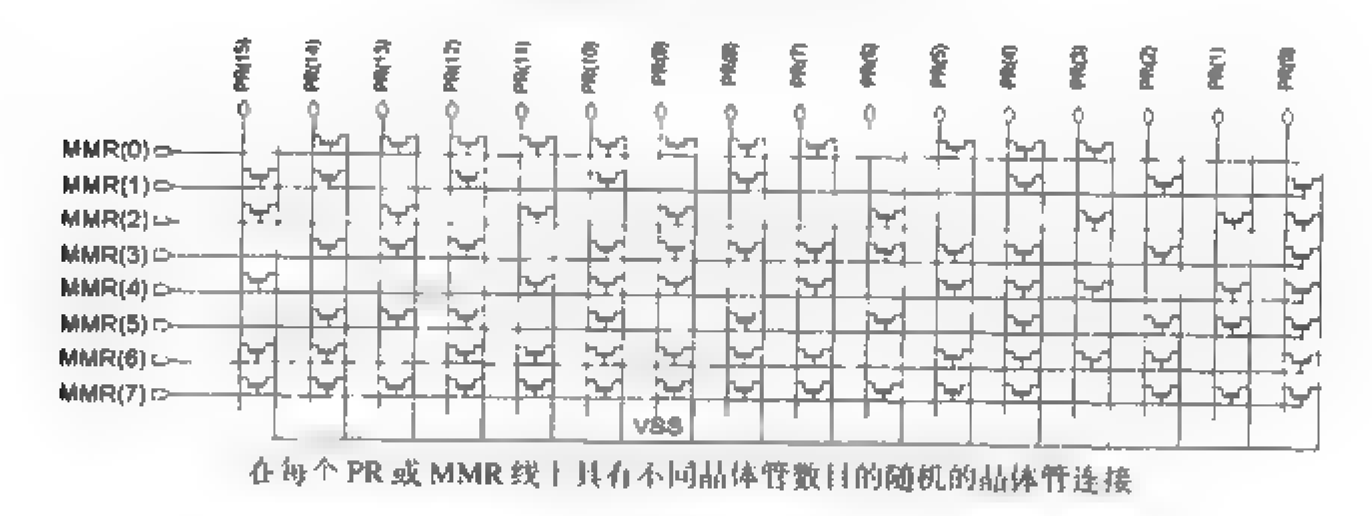
- 直到项目进行的后期，“片上”软件设计工程师都可以优化他们的代码。
- 如果合理地布局和实现，设计可以“一次性构建即可保证正确无误”，并且应该不会出现DRC和LVS错误。
- 一旦通孔编程技术发展起来，即使基本版图随之有一系列的变革，通孔也可以重用。只要在新结构中重新指定其“合法位置”就行。

● 有效的基本版图设计和编程方案应该包括这样一项技术：通过减少编程设计中通孔的数量使器件连通性最小化，以达到降低功耗、提高速度的目的。

图9.3说明了在MOSAID内使用的小型解码器通孔可编程性的例子。



在每个PR或MMR线上具有相同晶体管数目的固定对称结构



在每个PR或MMR线上具有不同晶体管数目的随机的晶体管连接

图9.3　解码器结构

9.1.3　测试Pad和探测Pad

测试Pad和探测Pad是一种辅助结构，它在设计中实现，以使产品工程师能够检查物理芯片的内部工作情况。

测试Pad指的是可以用做绑定的一种金属Pad；探测Pad比测试Pad小得多，是为专用的测量设备设计的。

测试Pad被规定仅在晶圆上有用，在最后的产品中无需绑定，在原型验证或评估阶段，通过钝化将这些Pad裸露在外，在封装前再用一层玻璃绝缘层覆盖起来。就ESD而言，它们比常规的Pad受到的保护要少，因为测试过后它们就不会和封装的引脚相连了。

一个测试Pad广泛应用的实例就是，在DRAM设计中有许多内部生成的、为内部电路供电的内部电源。测试Pad是普通大小的Pad，被连接到内部电源如VPP、VBB、VCP、VBLP

上。在原型阶段，测试Pad用来检测内部电源电路是否在起作用。在产品阶段，测试Pad也可以用于产品测试，以评估存储器的可靠性和工艺特征。

而探测Pad则广泛用于预生产调试和评估阶段。产品工程师使用探测Pad检查那些尚未投入使用的芯片的内部节点的功能和时序。

探测Pad可以放置在芯片上的任何地方，因为只有当芯片尚未受到任何保护时才使用它们（见图9.4），探测Pad的大小取决于所使用的设备。

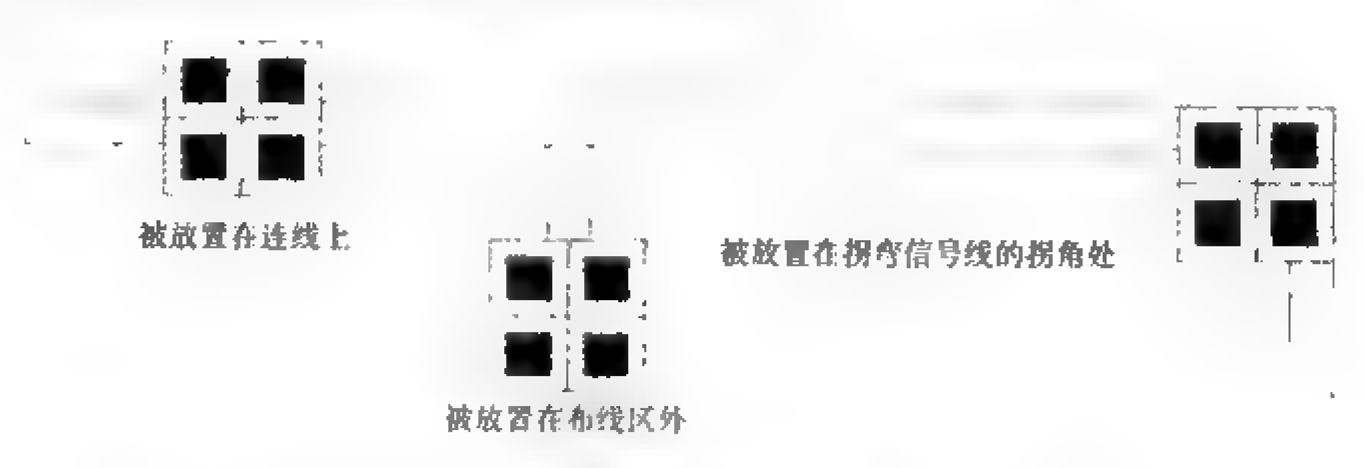


图9.4 探测Pad的例子

以前，并没有明确地实现探测Pad，因为对当时使用的探针来说，单个通孔已足够大。而随着分层尺寸的减小，单个通孔对物理探针来说已变得太小，通常面积在1平方微米内的探测Pad才足够。

应该在电路设计工程师的指导下完成探测Pad。尽管所有的布线节点几乎都应该是可探测的，但无疑应选择像时钟信号和数据通道信号等关键信号作为探测Pad。

另外，在文档里详细标注出候选探测Pad是很重要的，那样就很容易找到它们！

9.2 面向未知修改的规划

在设计的不同阶段，版图工程师要应对如下这些挑战：设计规则的修改、电路的修改以及软件工具中存在的bug等。最好的办法就是预防，但是很多时候我们却不知道会发生什么。在DRAM芯片中，对于主要的存储器分层，即使只对设计规则做少许的修改，原来规划好的结构都可能不再有效。因此，必须修改整个核心版图和设计。这样的工艺修改对ASIC设计版图的影响要小得多，因为ASIC的工艺相对更简单，而且大多数都使用了布局布线工具来完成版图设计。

面向未知修改的规划乍一听起来似乎有些奇怪，但是凭借电路设计工程师的经验，并考虑到设计中的新颖部分，我们往往能够估计出一个设计可能出现的修改。一般来说，在芯片级规划的时候，大多数的值只是“尝试”（wet finger）值。就此而言，即使用最先进的工具进行版图规划，都没有办法预见最终结构、最终的门数或者信号总数。最好的解决办法是在规划芯片版图时考虑到修改，并根据个人的经验预测修改的类型和数量。

举个例子，如果一个设计中有约50%的电路是全新的，那么我们可以开始对这50%到60%的可能的变化做规划。带着这种想法，就应该预留一些备用逻辑、备用面积、备用信号等，以应付设计的发展和变化。这样一来，设计成功的几率就会大增。

相反，若没有对可能变化做规划，初始设计的尺寸和进度将会持续增加到威胁项目的程度。老实说："在版图设计中任何事情都有可能"，只有想像力才会是一种限制，但是"面市时间"越来越成为实际的限制条件。有一个能够适应任何阶段所做修改的规划，它可以使项目进度的变化最小，并几乎不改变芯片尺寸，这是今天所有项目主管的梦想。因为错误是不可避免的，因此在开始设计各部分之前为修改做好规划，那么这个梦想就能实现。

我们可以使用一个称为ECO（Engineering Change Order）的正式步骤来处理最后时刻的修改（将在下一节中讲述）。在选择和使用这个方法之前，还有许多其他方法可以帮助我们处理这些突发事件。本节列出了其中的一些方法。

9.2.1 接触孔和通孔例化单元

我们在第3章中详细介绍了层次化设计的概念。使用层次化的方法有很多好处，但谈到面向修改的规划时，特别是应对大规模变化时，层次化设计的应用是减少数据管理问题的关键。

本节致力于讨论接触孔和通孔单元的优点，因为使用它们有很多好处。布局布线工具使用它们来进行连接，但是早在这些工具出现之前，人们就已经考虑到运用接触孔和通孔单元了。

具体的想法是为各种类型的接触孔开发一个单元库。当只具有三到五种类型的接触孔时，这种做法是有效的，尤其在DRAM存储器的设计中，这种方法确实提高了版图设计效率。DRAM存储器有许多多晶分层，而且还有多套设计规则，所以有大量的接触孔和通孔单元，因此在DRAM存储器设计中，拥有一个有24到36个单元的单元库是常见的。

在过去，接触孔单元只包含一个单一多边形：即只有一个接触孔分层。现在我们打算使用的方法（在MOSAID中是一个标准步骤）是创建一个包含三个相关层的接触孔单元。例如，在有源区和金属1之间的接触孔，将会是一个包含了接触孔分层、有源区以及覆盖接触孔的金属层1这三个分层的单元。该接触孔单元的版图完全符合设计规则，以至于该接触孔"单元"没有DRC错误。

这种方法的优点如下所述：

- 无需花费太多精力就可以全面实现诸如"金属端交叠"规则等高级规则。
- 这些单元没有DRC错误，所以其他的设计工程师就可以设计"一次性构建即可保证正确无误"的版图了。因为对ASIC设计工程师来说，有标准单元库，接触孔单元自然也就正确无误了。
- 对于拥有多套设计规则的每一款芯片来说，定义命名规范可以帮助版图设计工程师使用合适的单元。而且他们无需记住不同的接触孔设计规则，仅需要记住他们正设计的那一块就行了。

- 单元名可以从一个项目移植到另一个项目，不需要记住大量的设计规则。
- 由于接触孔单元边界的接触以及交叠都很充分，因此接触孔单元易于连接。
- 最大的优势在于可以节约时间。在“高风险的尖端技术”，比如存储器设计中，当工程师开始设计芯片版图时，工艺也许还没有被完全定义好。在笔者的经历中，有很多次都是用这种方法才做到按期交付。在许多情况下，我们不得不修改接触孔尺寸、交叠分层规则，或者仅接触孔分层。在正常情况下，所有这些修改都需要用某种CAD软件来处理，但是在它们被应用于整个芯片之前，必须对所做的修改加以检验并修正错误，而若利用接触孔单元，经验证明，通常只需要大约五分钟的时间，就可以成功地完成整个芯片的修改。

由于发现这一特性非常有效，所以我们决定把它应用于整个设计过程中。而且我们设法使提供晶体管级自动化工具的供应商们确信这一有效特性，并将包含接触孔单元作为它们工具的特性。在图9.5中可以看到一些接触孔单元的实例，包括所有的交叠分层。

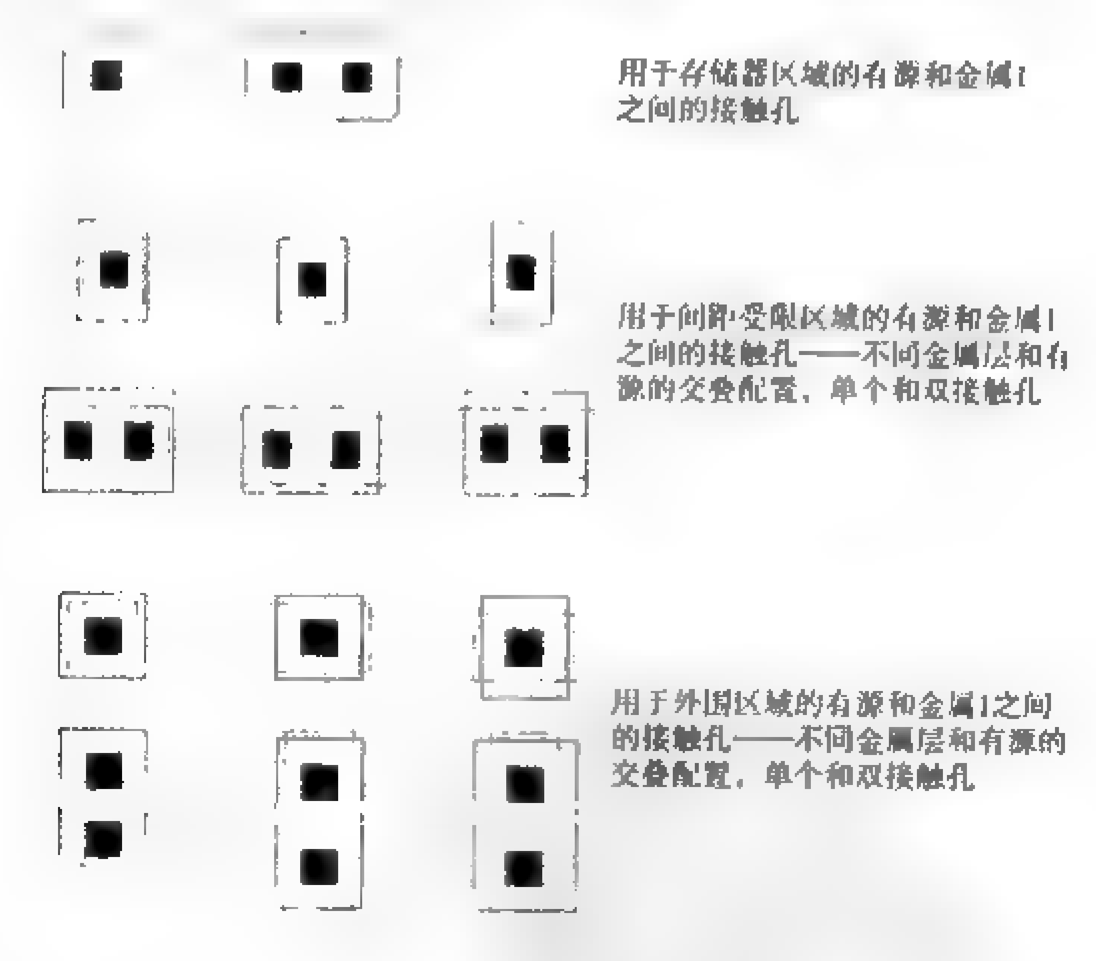


图9.5 接触孔示例

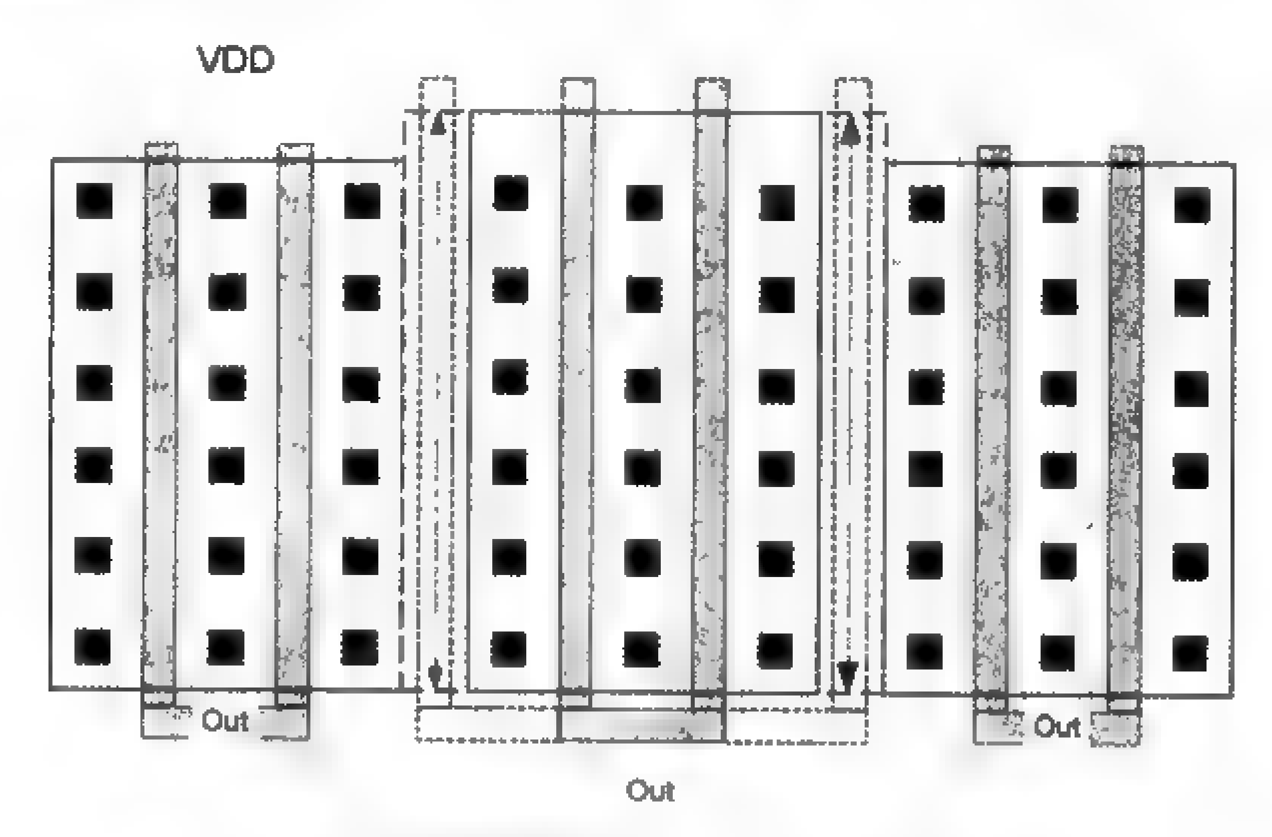
9.2.2 最小设计规则

另一种适应修改并防止修改对版图设计产生较大影响的方法是，避免总是使用最小设计规则或者最小面积的设计方法。如果已经知道正在进行的设计是全新的或极有可能被修改，那么这种方法特别有用。

考虑一下模拟电路设计的情况，设计工程师通常一开始可能并不知道一个模块的内部连线方式。在这种情况下，他/她设计中的最终尺寸将会随着从版图参数提取中所得的最终结果的不同而不同。

版图设计工程师如何才能避免移动接触孔、器件和单元以适应预期的修改呢？在这一节中，我们将概述一种解决方案。

前面讲过，版图设计最重要的步骤之一就是，设法尽可能地使用源/漏共用有源区来减小面积。如果出现"预期的"修改，就应该用这种方法来构建版图，以使它可以适应有限的修改。图9.6举例说明了这一点。



通过增加两个有源和两个栅极多晶，中心晶体管的尺寸可以增加到两个指状，而对于单元或芯片的面积不会有任何影响

图9.6　驱动大小调整

图中，在空沟道的一侧连接的是电源，而另一侧是输出。可能发生修改的是一个晶体管的输出。

图中所用的方法是，将有源区隔开能够容纳一个指状结构的栅极多晶的间距，该间距也许不是最小的，但允许灵活地修改晶体管的尺寸。

9.2.3　备用逻辑和备用线

即使使用最先进的设计工具并聘用经验丰富的专家，芯片也总是会有许多无用区域，并且这些区域的位置相当好，这使得采用备用逻辑门和备用线的想法是值得一试的。除非芯片的网表为展平的，并且设计组采用自动化布局布线工具，否则，任何设计最终总有许多未使用的空间。

设计的最后时刻的修改往往需要花费大量的精力，而掩模层的修改只不过是修改整个设计的一小部分，许多饱受修改之苦的设计工程师已经接受了在交付（tape-out）之前在芯片中放置一些不用的或者备用的门和信号线的设计方法。

其中一个简单的方法是设计一个包含一组普通逻辑功能的逻辑模块单元，并把它放在芯片设计的任何空余空间里。由于可能会失效的部分不尽相同，不同芯片中使用的逻辑门的数量和类型也不同。模块的设计很大程度上以金属层为参照，因此使用最少的层就能完成晶体管的连接和重构。在这种情况下，逻辑门起初是没有用的，这不会影响芯片的正常操作。

运用连接芯片不同区域之间的备用线并经由备用逻辑组传递，少数错误的修改就变得更加容易了。与失去市场相比，额外增加的线对芯片的尺寸影响就不是很大。

图9.7说明了这种在便于连接备用线的空余空间里插入备用逻辑门的想法。应将备用线布在每个通道中，并在芯片的完成阶段，除去要求布线的信号外，一般应备有最多为5%的备用信号。在设计的原型阶段，一个好的做法是每个通道至少含有两条备用线。备用线方向与通道的金属方向一致，并完全可以从芯片的一端连接到另一端。若使用大量的局部备用结构，那么可以实现全局的修改，而不影响芯片尺寸和进度了。

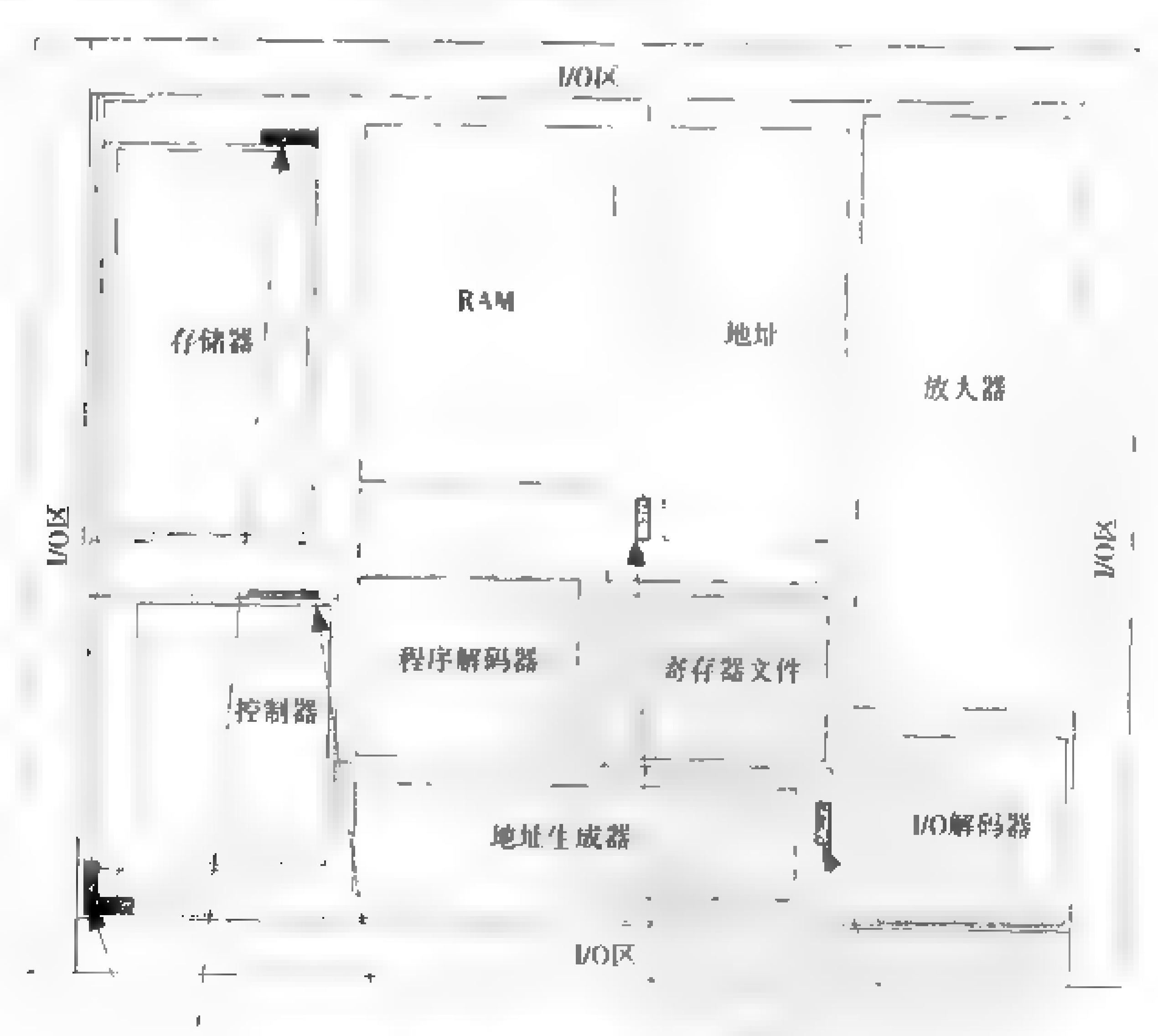


图9.7 备用逻辑和备用线

9.3 ECO

作为一个版图设计师，了解ECO（Engineering change order）是不可缺少的一部分。当一个设计接近完成时，错误的验证和修改就变成了一个非常重要的问题。采用诸如ECO流程这样的可控方法来完成修改，可以确保从各种层次的结构中检查到错误，不仅如此，通过检查一个模块中可能出现的改动，可以防止类似错误出现在其他模块的接口。

ECO是一个在设计项目的最后阶段使用的正式方法，它可以在不影响项目进度的情况下仔细地完成修改。若使用上述所有的方法，大多数修改对版图设计所产生的影响会很小。

就完成ECO这一步骤而言，由于有许多种设计流程和项目管理方式，因此就有许多种定义它的方式。

从理论或者管理角度上来看，ECO步骤如下所示：

1. 掌握整个芯片仿真和验证的进度，就可以确定设计的完成日期。
2. 这时候版图设计可能处在完成主要模块的最后阶段，顶层的布线也完成了90%。
3. 从这里开始的任何修改都被认为是ECO，从技术角度上看，有必要检查一下ECO，这可以确保它的有效性、决定它的重要性，当然，也要估计ECO对项目进度的影响。
4. 项目主管或项目经理必须检查ECO并决定是否要实行。如果错误较少或者可以在新版本的芯片中改正，则不需采用ECO。另一个评估ECO的标准是，它是否会影响到市场或消费者，或者说，芯片推迟进入市场的影响如何。
5. 如果要执行ECO并按时完成，则必须要用捷径，那样就可能要使用特殊的预防措施来确保能够正确地完成ECO。比如有第二个人来检查执行过程，或者采用产品分阶段发布的方式。

在ASIC设计流程中，可以使用布局布线工具来生成芯片的版图，该版图工具内置了ECO流程。

以下所述的ECO过程从概念上来说是简明易懂的，但实际上并不总是如此：

1. 检查新的网表，生成新的修改方法。
2. 布局布线工具可以删除网表中已不再存在的单元，并尝试加入新单元。在某些情况下，如果设计非常紧凑，没有太多的空余空间，那么新单元可能会被放置在离先前被删掉的单元很远的地方，再重新对单元进行布局。由此带来的问题是，在很多情况下，最初的布局是为电源、电迁移、RC进行过优化的，而ECO可能会改变这一情形
3. 布线工具可以取消局部布线，并尝试进行新的连接。现在大部分的流行布线工具可以处理99%的情况。而剩余的不能由布线工具完成的1%的工作，则可以通过人的观察以及手工编辑来完成。

4. 在完成布局布线后，需要进行版图的参数提取工作，来检查所做的修改是否满足时序目标，以及确保有关的电路未做太多的改动。如果没有达到要求，就需要反复进行额外的ECO流程。

9.4 适当的版图设计准则

前面我们介绍了大部分基础版图概念和设计方法，现在将它们概括成一个统一的列表。我们根据“自顶向下”地规划，而“自底向上”地实现这样一个基本流程来排列该列表。

9.4.1 芯片版图规划

- 从芯片设计工程师那里弄清楚芯片的体系结构，他们对可能的模块、功耗需求、信号组以及公司从未设计过的新领域会有一定的看法。
- 根据先前的项目，能够推断出一些模块的大小以及与之相关的信号数。我们需要这些信息来估计芯片和每一个模块可能出现的问题。

如果要设计一个新型模块，可能就必须引进新的流程和工具。这样，项目就需要额外的CAD支持来采用一些新技术。布局布线工具、压缩工具、新的验证工具、从其他供应商处购买工艺库以及IP模块等，这些只是新技术的少数几个例子，在规划中必须包括这些新技术。

- 获得并比较与其他已知工艺相关的工艺规则。具体包括以下内容：
  垂直连接图——特别是在有多种多晶类型的情况
  栅极的电阻和电容、金属导线、通孔和接触孔，以及每一层导电层的电迁移率。
- 评估布线分层，并为每一层确定布线栅格。出于简化的目的，判断在各个方向上将布线栅格统一成单一值的做法是否合理。
- 基于指状结构的尺寸计算模块尺寸（见图9.8）。根据设计的新颖程度以及设计工程师的设计经验增加不确定部分所占的比例。
- 检查封装并确定Pad的位置，特别是电源Pad、地址以及/或者数据总线的Pad。
- 确定电源的类型和栅格——基于电迁移或RC参数来确定电源线宽，以及为了满足允许的IR压降和阻抗，芯片在哪里需要特别的电源线连接。在一些先进的工艺中，电源线有它们自己的布线层。
- 估计信号线总数，针对意外事件构建通道，并为备用连线和备用逻辑做规划。

- 寻找重复结构，并设法重用其中任一可能的小块，以提高效率并减少修改带来的影响。
- 规划版图的层次结构和设计来尽可能使它们相互匹配（同样的层和命名）。即使在验证工具非常快的今天，排除错误所用的时间还与使用者熟悉错误并及时修正错误的能力有关。
- 通过将已完成的模块取代空盒模块并对内部模块的连接重新布线来定期更新版图。这样，当完成所有模块的设计时，芯片也就完成了。

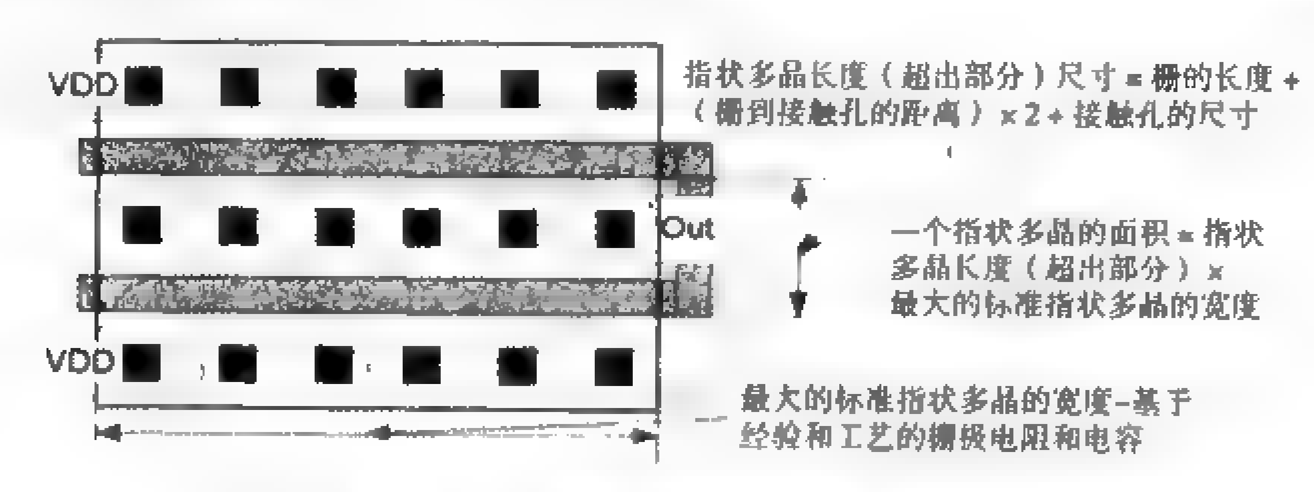


图9.8　版图规划的基本计算方案

9.4.2　模块

模块级版图设计准则如下：

- 从芯片级输入模块大小和端口位置，并设法使之固定不变。
- 定义直通信号，这样就可以减少整个芯片内部模块通道内的阻塞。
- 定义电源要求和每个模块内的电源网格。
- 定义关键路径设计，并将相关单元分成一组来使布线最少。
- 通过留有备用逻辑和备用连线来为修改做规划。
- 通过使用EDA工具来提高效率。当组元数达到100或者更多的时候，即使优秀版图设计工程师也会发现在完全遵守针对这些例化单元的所有设计约束的同时，还要很快完成工作是相当困难的，但是如果设计工程师使用计算机辅助布线而不是用手工连接，即使工具仅能完成80%的工作量，也会使设计更快、更准确。

9.4.3　单元

对于单元级设计，我们提供如下的总准则：

- 定义单元的体系结构——如标准、全定制、数据通道等，以及基于模块规划的边界规则
- 定义电源的类型和宽度。
- 定义特殊要求。例如，对称、相邻、关键路径。
- 根据速度、功耗以及布线要求（空隙率）来定义晶体管的类型。
- 和相邻单元一起验证单元的 DRC 规则。在较高层次上，这可以消除所有可能发现的错误。

第10章　版图设计的计算机辅助设计工具

10.1　概述

每年，在世界最大的IC CAD会议——著名的设计自动化年会上，各种提高设计效率的新产品或新设计方法都能让来自世界各地的VLSI设计工程师们眼花缭乱。

会议前，杂志上和电子邮件上的广告如潮水般涌来。在会议期间，产品介绍和展示一直都在进行。在此过程中，展示工具的市场销售人员充分展现工具的新选项和益处，诸如附加功能和特性、多种菜单选项，以及与使用人工方法或同类产品相比而言的所谓"最重要"的性价比。然而，这些展示中却没有谈及工具的基本原理，工具设计工程师的想法，工具提出了哪些新的设计理念，或者工具支持什么样的流程。

我们了解这些工具具有哪些按键，易于使用的特点，所要输入表格的值，但我们不了解使用这些工具的真正理由。在很多情况下，那些具有很大潜力的新工具最终没能成功，因为没有用一种正确的方法将它们介绍给用户。

事实上，那些将工具卖给对口用户的供应商们往往会更为成功，即使其他供应商拥有更好的工具。在过去的15年中，曾有许多公司试图开发一种工具，通过一个按键就可以省去版图设计工程师的工作，然而今天它们大多已经销声匿迹。

举个例子，在Motorola公司工作时，我们收到了一个新的版图设计工具的试用版。这个介绍工具的应用工程师在开始演示时这样介绍到："这个工具可以省去版图设计工程师的工作，提供工程师所需要的一切！"我想我不用解释，不管这个软件有多么成功，这个错误的广告词就足以使一个功能强大的工具遭遇失败。

再比如，有某种工具，在时序方面的优势明显并且功能很强，而且它的开发者还意识到了把版图和电路仿真数据库整合起来所能带来的优越性。这样的工具，若再加上良好的客户关系和一定的市场根基，是完全有可能发展成为市场的领军产品的。然而，开发者却错误地以损害其他用户为代价，只迎合部分用户的需求，最终导致了该产品的失败。很明显，本书中将不会公开这个工具的名称。

在本章中，我们将介绍我们所了解的各系列工具所提出的设计理念，以及使用这些工具所需的相关方法。工具和供应商的详细名单请参阅集成系统设计杂志网站，那里有最新的供应商名单。

下面我们将概述用于当今CMOS VLSI 版图设计的基本工具类型。后面将依据流程以及工具所处的环境和应用领域，展开介绍这些工具的特性和典型用法。

各种类型的工具根据其功能不同可分为三类，如图10.1所示

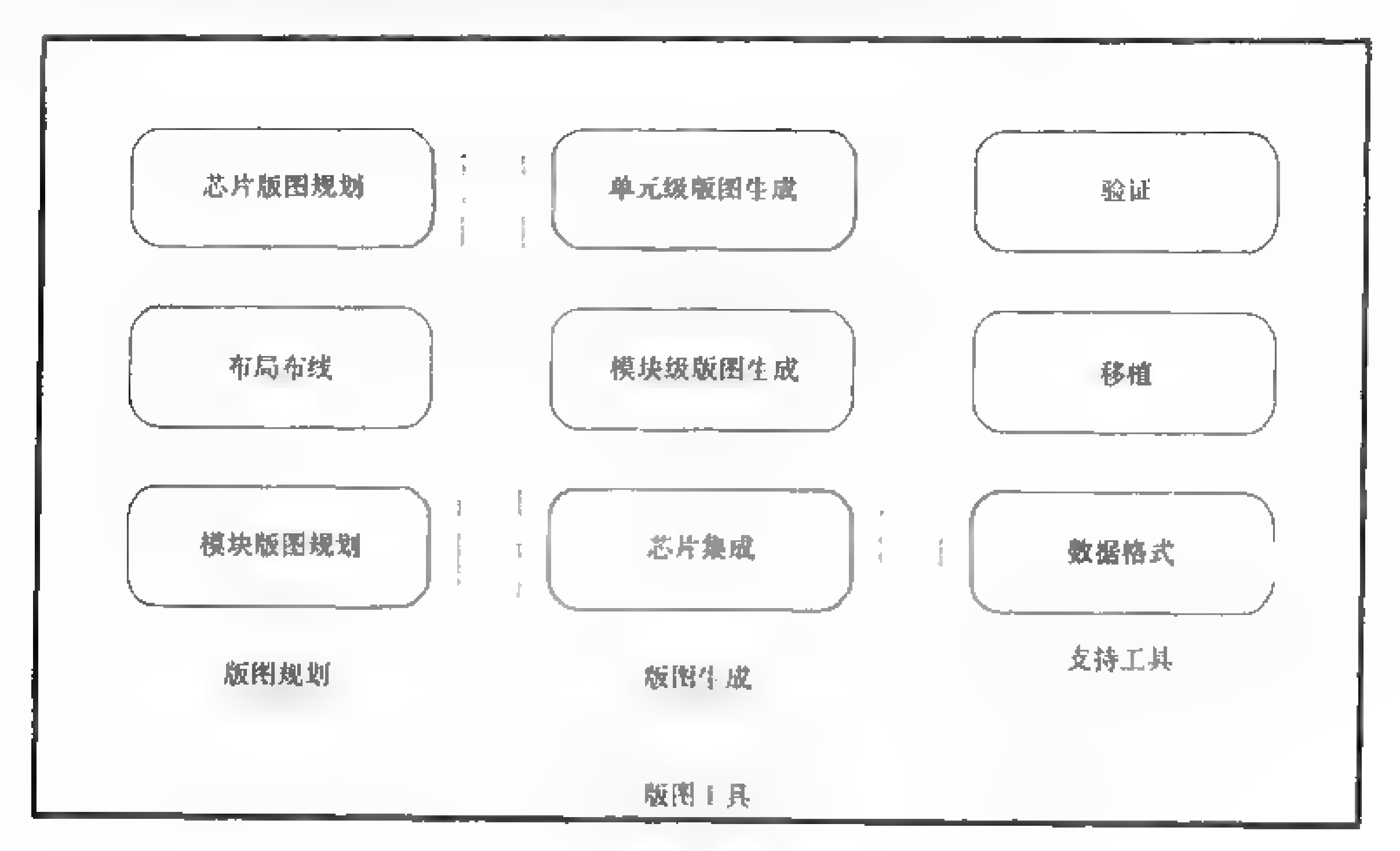


图10.1 版图工具按功能分组

1. 规划工具包括：

- **版图规划工具**：版图规划工具通过协调布局和布线引擎来生成版图规划。这类工具针对版图设计提出了新的设计理念和挑战，因而受到广泛的讨论。如果被适当地应用，版图规划工具可以提供自顶向下的设计方法以及“一次性构建即可保证正确无误”的版图，从而缩短产品面市时间。
- **布局工具**：布局工具利用物理和逻辑约束条件来优化单元或器件的布局。布局工具通常与特定的布线工具共同工作，因此使用同一供应商提供的布局工具和布线工具显得格外重要。这是因为这两种工具需在一起工作来满足版图的约束条件，并且充分利用为两个工具所知的特性和信息。

  为了更好地理解这个话题，我们将对与设计的特定层次有关的布局工具的特性和限制展开讨论，一旦理解了这一点，将很容易理解为什么布局工具和布线工具协同工作的效果更好。举一个简单的例子，一个通道布线工具需要布线的通道，而布局工具必须提供这些通道，如果没有这种功能，即使是世界上最好的布局工具，也是废品。

就布局工具类型而言，按层次分，可分为三类，每类都有不同的特性和要求：

在单元内——晶体管或单元的布局。

在模块内——单元、模块，或单元模块的混合布局

芯片级——应用版图规划工具进行的模块级布局。

- **布线工具**：布线工具是第一个被广泛使用的自动化工具。布线工具可以提高版图互连速度。最初的布线工具仅能做芯片级布线，而现在它们已能处理单元级布线了。当连线过于复杂、使用人工难以完成或效率过低时，布线工具就变得必不可少了。

2. 版图生成工具包括：

- **版图编辑工具**：利用图形用户界面，Polygon pusher 工具或版图编辑工具被用来生成多边形和线形。这些工具中有些功能非常完善，甚至可能包括布局布线功能。
- **符号编辑工具**：符号版图编辑工具有着与Polygon pusher 工具相同的用户界面。它可以根据工具内置的编码或数学算法，以符号的形式生成版图。这种方法的优点是，编码时可以把工艺设计规则设成参数，这样，就很容易针对不同工艺生成版图了。
- **器件生成工具**：器件生成工具是用来生成版图器件的，例如晶体管、通孔阵列或逻辑门。它们的特别之处在于具有丰富的图形用户界面和高度发展的宏语言。在某些情况下，器件生成工具是对版图编辑工具或其他工具的拓展。但是，若没有布局布线工具，器件生成工具对生产率的提高是非常有限的。
- **压缩工具**：它可以自动优化已有的版图，通常用做先进的版图编辑工具或符号版图工具的拓展。压缩工具可以缩放宽度和多边形之间的间距，为的是在符合工艺设计规则的情况下，实现版图最小化。
- **硅编译工具**：硅编译工具通过生成晶体管、叶单元以及使用基于标准结构的叶单元的结构，来自动生成版图。由于硅编译工具用来处理大量的结构，往往没有图形用户交互界面，通常是由具有丰富软件经验的人来开发它。

3. 版图支持工具包括：

- **版图验证工具**：版图验证工具对完整的版图进行一系列的检查，包括设计规则检查（DRC）、版图电路图对比检查（LVS）、电学规则检查（ERC）、版图参数提取（LPE）以及光学近似工具。
- **绘图工具（Plotters）**：绘图工具一般不是软件工具，但为了产生有意义的、可读的版图硬拷贝，也会使用这些专为处理版图而设计的软件。
- **转换工具（Converters）**：版图转换工具常称为移植工具。与压缩工具类似，转换工具用来转换和压缩已有设计，将其重定义到具有不同设计规则的工艺中去。

在本章的最后，将讨论版图数据库的数据格式。

10.2 版图规划工具

无论面临什么样的版图设计任务，都需要版图规划这一步骤，但通常版图规划都是从芯片级到模块级的。这一方法学所隐含的设计思想是自底向上地构建所有内容，而利用顶层版图规划来定义模块接口；并且当所设计部分完成和经过验证后，利用版图规划来协调这些接口的所有更新。

10.2.1 芯片版图规划工具

版图规划是这样一个过程，它确定应该将哪些结构放置在一起，并给它们分配相应的空间，以使尽量满足有效空间（芯片成本）、所要求的性能、对每个模块之间进行无缝接合的希望等这些可能相互冲突的目标。

对大多数芯片来说，最小化的设计也是最高性能的设计。因此，面积和速度是两个并行的重要特征。面积小的模块或芯片，连线更短、布线更少，因此点对点的信号通道更快捷，甚至布局布线也更快速、更稳定。

采用版图规划的方法，应该能使设计的面积最小化，因为这种方法把版图设计工程师在划分电路上的经验和版图规划工具的优化算法结合起来，从而可以有效地规划设计。

图10.2给出了设计过程开始时版图规划的不同组元。

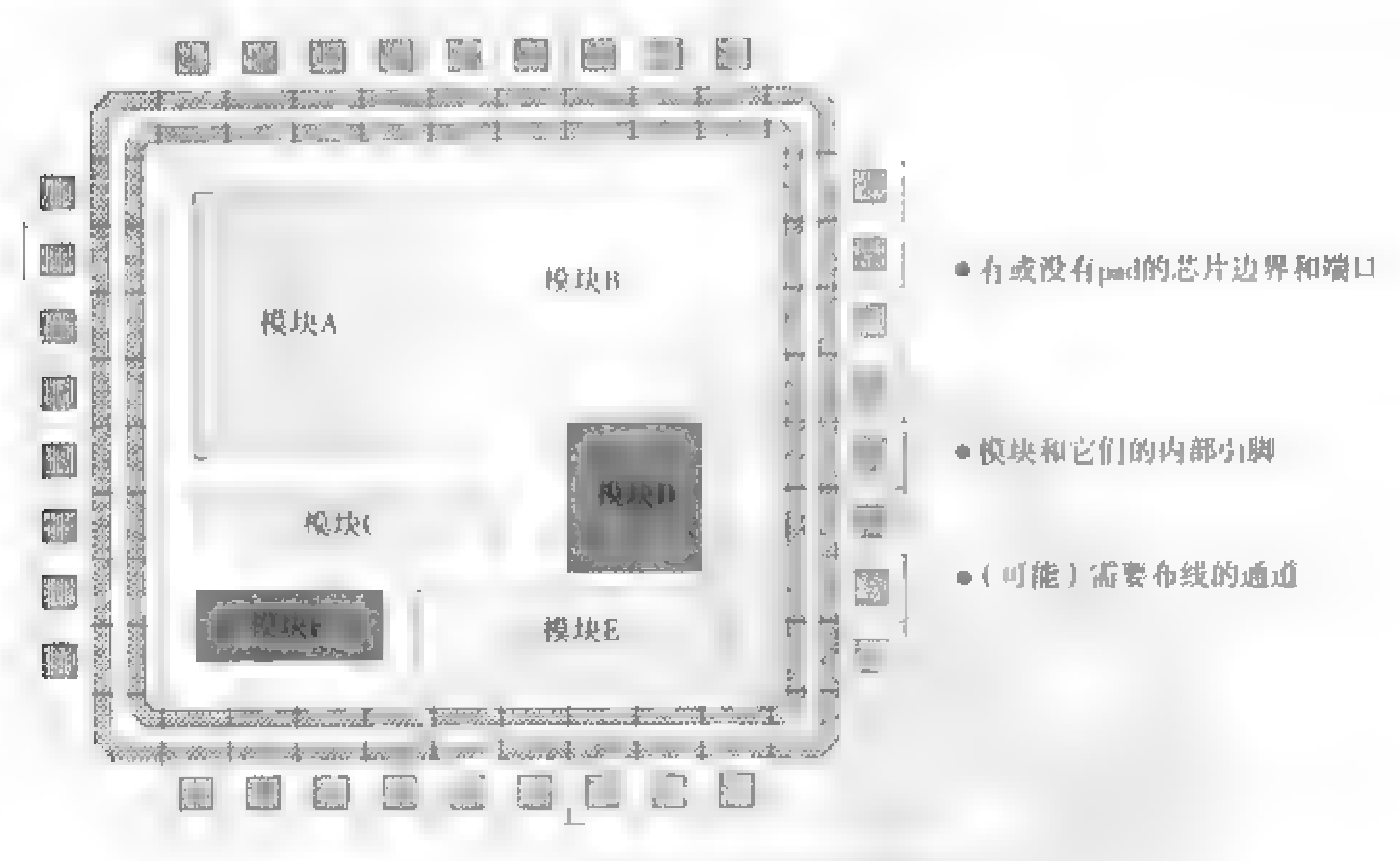


图10.2 一个简单芯片版图规划的组元

为了理解版图规划的作用以及版图规划工具如何帮助设计工程师快速、准确地获得结果，掌握版图规划工具的设计理念和实际功能是很重要的。版图规划工具能完成如下任务：

- 理解一个设计的不同分区或模块。
- 理解每个模块的关键特征值：大小、长宽比、引脚
- 动态地显示模块之间的连接以及和 pad 之间的连接。
- 依据布线层数为布线分配空间。
- 依据约束条件放置顶层端口。
- 放置每一个模块，并依据总的连接要求和模块间布信号线的可行性，来优化每个模块的引脚位置。
- 允许用户对版图规划进行增量修改或完全重新规划。

图 10.3 强调了在模块可以被芯片级版图规划调用之前，所需的关于模块的重要信息。

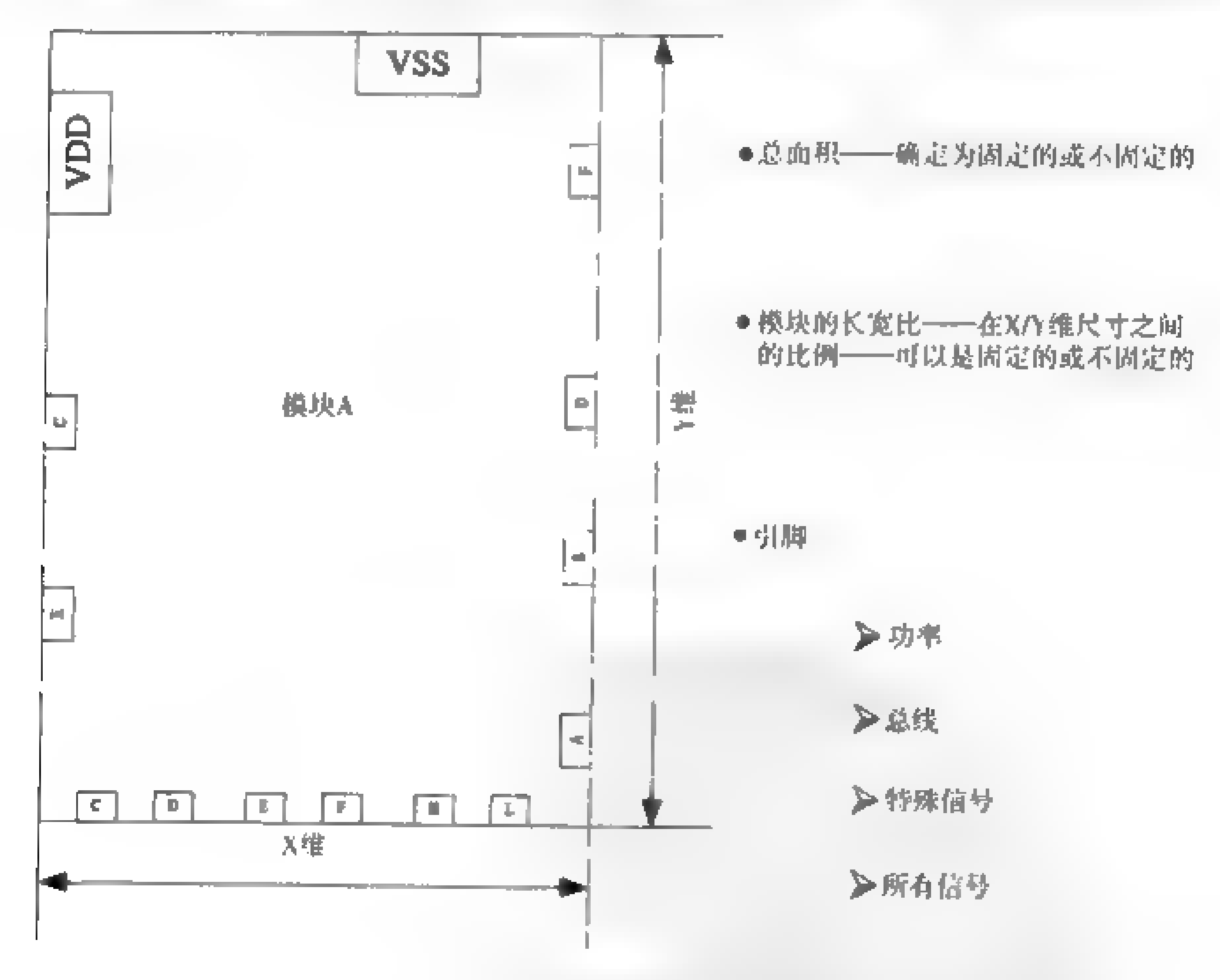


图 10.3　版图规划所需的模块信息

图 10.4 列出了有效利用版图规划工具的步骤。下面是一些运用版图规划工具时应记住的提示：

- 首先，所有顶层pad应连接到单独的层上。这种约束会影响布线工具优化和pad之间连线的能力。在某些情况下，pad可以被分配到两层上，这样就会给布线工具更大的布线自由。在有许多可用布线层的（多至10层以上）情况下，最好由版图设计工程师手工将pad分配到布线层上，因为可能有些层已被指定给一些必须被分配的端口，如电源端口和时钟端口。
- 与先前情况相似，模块引脚有时也只能由手工分配给合适的层。
- 插入备用逻辑和备用信号是需要技巧的。一般而言，设计网表中并不包含备用元件，因此这些元件也只能用手工插入。使用工具的人应掌握如何修改网表，他既可以引入“虚构的”线和引脚，也可以为备用逻辑扩大模块或添加虚构模块。为了确保备用线紧靠备用逻辑，以更容易地使用它们，对备用逻辑进行适当布局是十分必要的。
- 应当巧妙地放置那些可以改变尺寸的模块，这样才能有额外的空间用于模块的尺寸增加。

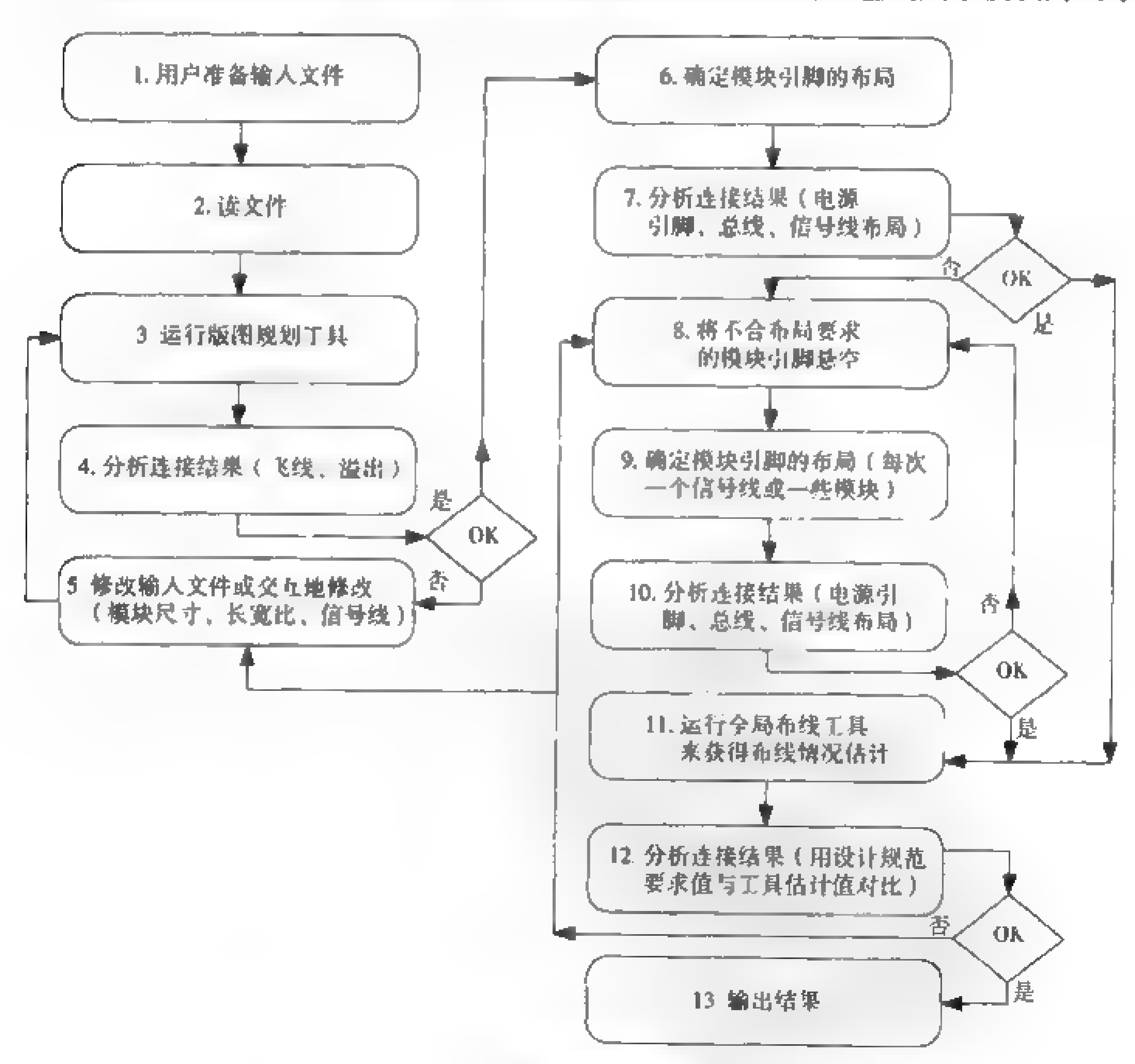


图 10.4　版图规划步骤

10.2.2　模块版图规划工具

在模块级，版图规划步骤与芯片级版图规划步骤基本上一致，只有很小的差别。此时的版图规划工具处理的对象不再是模块，而是依据设计功能和连接而定义的一组单元。在这种情况下，设计更加成熟，因为此时设计中的单元和连接关系被更好地定义了，并且具有未知内容的空模块会更少。

模块级版图规划与芯片级版图规划相比较而言，既有相同点，也有不同点：

- 模块级可能面对数千个单元，而芯片级面对的模块数量通常少于 20 个。
- 尽管模块级有更多的端口，但端口的处理是相似的。
- 模块级引脚数达上万个，而且连接的复杂性也比芯片级高。
- 如果工艺的金属层超过三层，版图规划将不需要通道——这个规划看起来像一片由单元构成的海洋。
- 如果有一些模块是购买的核或硬核，版图规划工具将在其余的逻辑中优化这些模块的布局。

版图规划工具必须依据设计需求或设计使用的整个流程输出信息。图 10.5 说明了版图规划工具提供的数据格式，不同的输出格式取决于其应用目标。

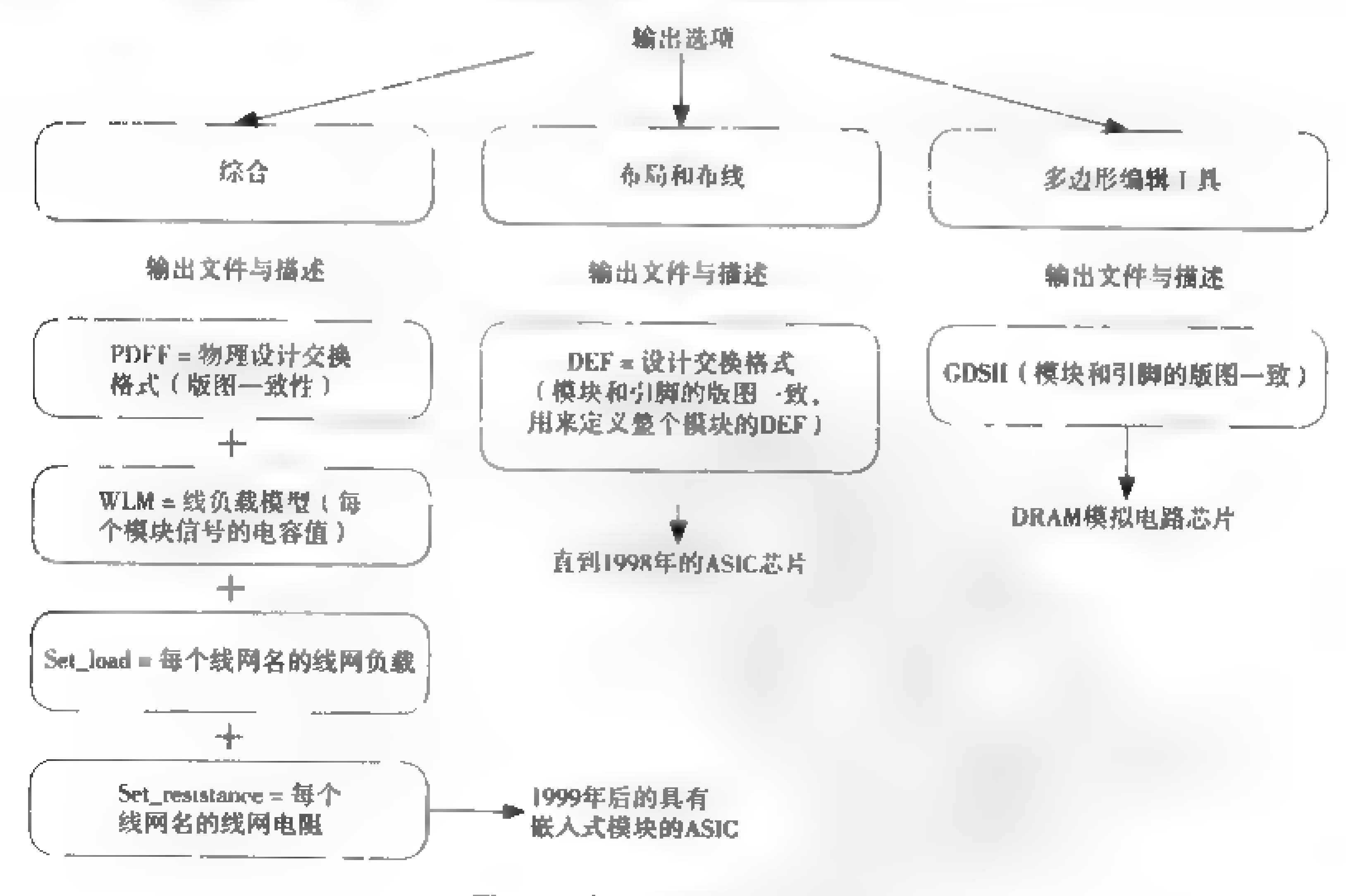


图 10.5　版图规划输出文件格式

10.3 版图生成工具

版图生成工具是目前市场上版图设计工具的主要部分。不同的抽象层次将使用不同的版图生成工具，共分成三类：基于单元的版图生成工具、基于晶体管的版图生成工具，以及芯片集成工具，本节将分别讨论这些工具。

10.3.1 单元级版图生成工具

单元级版图生成工具通过多边形来形成晶体管并完成晶体管间的连接。

Polygon pusher工具。最流行的单元级版图生成工具仍是Polygon pusher工具，或称为基本的“版图编辑工具”。理论上，Polygon pusher工具可以用来设计晶体管级及之上的所有类型的完整芯片。然而，对于模块或芯片的集成，它却并不实用。

Polygon pusher工具不仅可以用来生成包括10到100个晶体管的低层次单元，还可以用来绘制VLSI工艺所需的所有分层。这种工具非常成熟，一般不需要太多维护。然而，某些公司仍花了大量时间去改进那些通过重复使用宏单元来生成版图的基本工具。

Polygon pusher工具为高质量的全定制版图设计提供了开发环境，但它们完全由手工操作，速度较慢。长期以来，人们一直致力于改善其开发环境，比如试图给设计过程的某些阶段加入自动化操作：如N阱生成、接触孔单元、晶体管生成、端口宏单元等。

几乎所有大型IC CAD供应厂商都提供版图编辑工具，因此，根据这些工具的自动化附件的有效性来评估它们是非常重要的。除了少许特例，这些工具大多运行在UNIX机器上，但也开始向PC机转移。

版图编辑工具的具体应用包括以下几个方面：

- 全定制的VLSI版图设计，对这种类型的设计来说，尺寸是最重要的因素。DRAM、SRAM、PLA、Pad输出缓冲器、输入保护器件等中的那些间距受限的版图，事实上仍需用Polygon pusher工具来完成。
- 模拟电路设计和特殊工艺单元或keys的设计。

版图编辑工具的特性如下：

- 用户可以用它们来做任何类型的版图设计。
- 无论对任何工艺，它们的设置都相当容易，并且在大多数情况下不需要定制。
- 入门仅需简单的培训。例如，高级版图设计工程师在设计模拟电路时只需要理解相关的版图设计概念，而不需要过多了解工具的特性。
- 长期以来，许多公司围绕它们的版图编辑工具已经形成了整套的方法，这就是为什么其他独立的自动化工具至今还不能取代它们的原因。

- 版图生成工具依赖于工艺，因此过去人们一直认为应该以100%全定制的方式完成所有设计，而ASIC流程和布局布线改变了人们在版图设计方面的观念。
- 在大公司内部，对于要求CAD小组去开发和维护设计环境的特定任务，往往有许多内部工具。很多时候行政利益与发展和效率是背道而驰的。而新兴公司却不必考虑这些问题，因此他们很容易接受新工具。
- 这些工具与自动化工具相比，其速度较慢，这也正是很少单独使用它们的原因。它们越来越多地和复杂的器件生成工具、晶体管布局布线工具、符号编辑工具、压缩工具以及布线工具一起混合使用。所有拥有大量客户的供应商们正努力通过添加某种类型的自动化操作功能来改进这些工具。

器件生成工具。带有布线工具的器件生成工具几乎可以用于实现各种复杂的版图设计，然而大多数还是针对那些应用于不同领域的低层次或晶体管级版图设计。

这个工具可以用来快速、高效地生成低层次单元，并且/或完成标准单元库，此时每个单元通常被限制为10到100个晶体管，并且可以绘制VLSI工艺所要求的所有分层。

安装和维护此类工具需要广泛的知识。器件生成工具将版图设计自动化和全定制设计结合起来生成单元。

如今，现有的高级器件生成工具是由电路图和网表驱动的，因此，生成的器件可以"一次性构建即可保证正确无误"。博学的版图设计工程师可以利用这个"增强型版图编辑工具"创造出奇迹。

器件生成工具的具体应用如下：

- 标准单元库——单元引脚分配、单元高度、相邻要求等符合标准，而这一点在版图设计中十分重要。
- 用于数据通道的单元——工具和设计要求必须得到保证并适应特定的设计。
- 质量要求和模拟电路要求和版图设计速度同等重要的情况。

器件生成工具的特性如下：

- 可以很快地生成版图，但对小公司来说价格很高。在我们撰写本书时，由于市场竞争的关系，它们的价格正不断下降，因此被认为是下一代基本的版图编辑工具。
- 这些工具需要由有经验的用户安装，但却可以被初学者使用。另外，对于某些特殊的版图需求，可能需要版图方面的专家以及宏单元，但这种情况仅占单元设计工作的十分之一。
- 与全定制的Polygon pusher工具相比，它们的速度更快，这也是为什么它们在过去五年中获得市场青睐的原因。然而，比起硅编译工具，它们的速度要慢
- 在设计过程中，这些工具为工艺移植和/或工艺修改提供了一个良好的环境——能够处理到两年的项目设计期内工艺发展所带来的问题

原则上，硅编译工具的目标正转向结构规则的和模块级的版图市场，而器件生成工具正准备代替旧的Polygon pusher工具。

单元布局工具。单元布局工具用来优化单个器件及分层的布局。在许多情况下，为了提供一个完整的版图环境，器件生成工具会和单元布局工具、单元布线工具结合起来使用。为了使布局更合理，使用该工具时需要了解相关单元结构。

单元布局工具有如下特性：

- 工具处理的器件和引脚数很少（数百个），但布局的优化却非常复杂，因为必须考虑到许多约束条件。
- 端口能被放置在所有边上。这种布局可以是固定的，也可以是“可变的”，因为若能获得一个更佳的版图，此布局是可以修改的。
- PMOS和NMOS晶体管的布局区域是可控的。例如，晶体管既可以放置在同一行的不同高度上，也可以放置在不同行的不同高度上。
- 依据特定的结构约束，可以自动将晶体管变成指状结构。
- 可以控制衬底和阱的连接。
- 可以输入已完成的版图，这一点对于以下情况十分有用：先用手工完成一个粗略的版图，然后交给自动化工具完成其余的工作。
- 与单元布线工具之间无缝接合。

在单元层次上，布局工具可以真正提高版图设计工程师的速度和生产率。可以说运用布局工具迈出了提高单元版图设计自动化程度的重要一步。在单元级版图设计层次上，运用版图布局工具的缺点很少，惟一的障碍是它们必须便于安装和使用，并且由于使用它们的人大多是那些熟练运用多边形编辑工具的设计工程师，因此它们需要良好的图形用户界面。过去，所有基于文本的工具都因此而失败了。

单元布线工具。针对单元级版图的布线工具通常十分简单，然而有些供应商通过提高单元级版图设计功能来满足芯片级集成要求，这样，就为它们的发展找准了位置。目前，市场上有几个非常好的单元布线工具。将器件生成工具、布局工具和布线工具集成在同一个环境下非常重要。如果单元布线工具是由网表驱动的，则可以加速单元的生成。功能最强的布线工具允许用户定义如下的约束条件：

- 等长的信号线——对于整个总线中的所有位，其信号线都具有完全相同的长度。
- 预定义的信号具有特定的线宽，或者交互地应用特殊需求。
- 电源布线约束。
- 每个连接的接触孔数目，以及/或源/漏的最少接触孔数目。
- 差分对的布线。

压缩工具。无论简单或复杂，几乎所有的版图都可以使用压缩工具，但压缩工具在晶体管级和单元级上应用最广，能用来压缩单元设计中晶体管的版图以及它们之间的连线。

对于版图设计工程师来说，使用此类工具的一种方法是，先完成粗略的版图设计，然后使用该工具对其进行优化，这是一种能够生成没有 DRC 错误的版图单元的快捷有效的方法

对于单元级版图，压缩工具的安装和维护要求设计工程师有渊博的知识 使用目前先进的压缩工具，同时使用由电路图或网表驱动的版图生成工具，就能产生最好的版图设计，因为该版图设计可以“一次性构建即可保证正确无误”，并且能通过 DRC 及 LVS 检查。当压缩工具处理符号版图数据时，其结果非常快，而且如果需要的话，它们可以添加高级结构，如连线中的 jog 结构。

压缩工具能用于设计的各种层次——从晶体管级版图设计到顶层布线，压缩工具是所有版图设计环境的一部分。

压缩工具的特性包括下面几个方面：

- 它们能快速生成大多数类型的版图，并且能在 DRC 错误产生前提前校正错误
- 只要专家确认安装正确，任何新手都可以使用该工具
- 压缩工具有时用于处理有限的工艺移植，以及/或处理设计过程中的工艺变化，比如有一个项目，耗时一到两年，期间工艺发展了，那么压缩工具可以用来处理较小的设计规则修改。
- 目前，压缩工具仅在展平式模式下或在层次结构的单层中运行。未来可能会出现一种按键式工具，能够压缩整个芯片来调整设计规则的改变

硅编译工具。硅编译工具可以用来设计各种复杂的单元 与压缩工具相似，大部分编译工具用于晶体管级或单元级版图设计时效果最好。硅编译工具通常用于生成低层次单元，以及/或生成限制在大约 10 到 40 个晶体管的标准单元，并且在绘制 VLSI 工艺所需的所有分层时非常快捷有效。

为了在一个可变的环境中发挥作用，硅编译工具的维护必须全面、专业、该工具能尽可能快地生成单元。当为特定的布局工具或布线工具定制单元时，该工具一天内就可生成一个标准单元库。

硅编译工具的具体应用如下：

- 标准单元库——单元引脚配置，单元高度，相邻需求等符合标准，而这一点对版图设计十分重要。
- 用于数据通道的单元——工具和设计要求必须得到保证并适应特定的设计。
- 在版图生成中，速度在任何时候都是最重要的因素。

硅编译工具有如下特性：

- 可以很快地生成版图，但对小公司来说，它的价格十分昂贵。例如，针对标准单元库的硅编译工具是如此昂贵，以至于能买得起的只有那些把库作为主要产品的公司。
- 只有那些在软件方面受过培训的人才能够使用该工具，例如软件工程师和/或具有丰富软件知识背景的设计工程师。而这些人可能只知道如何运行这种工具，却未必知道如何看版图和使用版图。我们希望本书可以帮助他们更多地了解在整个设计流程中版图生成部分的相关知识。
- 在大公司内部，对于要求CAD小组去开发和维护设计环境的特定任务，往往有许多内部工具。因此，使用标准格式来连接硅编译工具的输出和流程中的其他工具。但这样一来，硅编译工具所要求的特定信息可能丢失，使用这个工具的优势就不存在了。
- 与全定制的Polygon pusher工具相比，它们的速度更快，这使它们在过去五年中获得了市场的青睐。硅编译工具不仅可用于生成版图，还可用于工艺移植。
- 有些硅编译工具应用于特殊领域：RAM、ROM、PLA、I/O单元、标准单元、数据通道设计等。这些硅编译工具不要求大量的培训，因为它们是专为新手设计的。

尽管硅编译工具是包含器件生成工具、布局工具以及布线工具的一整套工具，但它并不能提供交互式编辑功能，这只有通过将三个工具结合起来使用才能够实现。

10.3.2 模块级版图生成工具

模块级版图设计通常并不考虑晶体管设计，而是考虑利用那些小型单元和宏单元，而它们是用10.3.1节中提到的那些工具构建的，在模块级版图设计中有三类工具：tiler工具用于规则结构，布局、布线工具用于不规则的布局和连接。

tiler工具。tiler工具用于自动生成诸如存储器、数据通道电路以及Pad框等规则的和重复的结构。其图形用户界面（GUI）一般都非常简单，它们既可以作为单独的工具来使用，也可以被集成在具有多种工具的设计环境中使用。

tiler工具是一个简单的布局工具，它可以解析一个预先定义好的或编程好的结构，并且可以将叶单元放置在已经定义好的阵列中。tiler工具可以简单地执行一系列版图指令（Tiling程序），却不能基于网表或约束条件来优化或重构设计。tiler工具最普通的形式为存储器编译工具的版图生成部分。

高级的tiler工具可以在它的Tiling语言里支持判定结构（例如，if-then-else结构）以及参数。这个特征使得tiler工具可以依据一系列参数来配置该工具，这也正是灵活的存储器编译工具的基础。例如，如果单元的选择依赖于某些特定的参数，那么这些参数就可以被编入Tiling程序中。用同样的方式可以选择和编辑缓冲器的强度。

下面是tiler工具的一些具体应用：

- RAMS、ROMS、PLA、I/O、DATAPATH 等
- 新一代移植工具利用tiler工具分解某个特殊设计的层次结构，然后将此信息提供给移植工具，进而自动生成叶单元的邻接约束条件。

tiler工具有如下特性：

- tiler工具可以用做快速面积估计工具。它可以快速生成一个有代表性的单元系列，来得到该芯片或模块的面积估计。如果再添加布线过程，并对其进行参数提取，就可以估计互连延迟，这一点在ASIC设计流程中尤其有价值。
- 只要专家确认安装无误，那么任何新手都可以使用该工具。
- tiler工具一般不贵。

模块布局工具。模块级布局和单元级布局大不一样，模块布局工具是ASIC设计流程中所使用的主要工具之一。它们通常称为布局布线工具中的布局部分。随着芯片规模和复杂程度的增加，它对于未来的版图设计是至关重要的。

单元级和模块级布局工具最大的不同之处在于网表的规模，后者可能包含成百上千个单元和信号，并有成千上万个端口信号进入模块。在一定时间内，这种规模的设计单凭人工是不可能完成的。实际上，用人工设计的模块所含单元数被限制在500个以内。

模块布局工具是一个能处理大规模设计的高级工具。该工具能利用复杂的算法来优化和重新优化单元的布局，使之满足时序约束、面积约束和可布线性约束。这并非是一项简单的任务，因为这些工具必须有效地处理这些折中。

理解布局工具所起作用的方式之一就是理解该工具的输入：

- 电路设计网表。该网表包含了将被布局的单元列表以及它们之间的逻辑连接关系。
- 对设计中每个单元的描述。该描述包含一些诸如尺寸、引脚、引脚位置、功耗以及时序特性等特征。
- 模块总的有效面积和端口的布局。在基于单元的布局中，布局工具需要所有有关的行信息，如大小、方向、通道限制（若有的话），以及黑盒或硬件宏单元的位置。
- 在门阵列情况下，允许放置单元的合法位置。
- 每个单元和整个设计的约束条件。如果需要使用诸如电源网格的估计、每行的IR压降或电迁移定律等高级特性，就需要输入这些条件。

模块布局工具的输出是设计的最初形式，在这里所有单元都被放置在特定位置上，并且整个设计已为布线做好准备。

基于时序约束的布局方法是一种可行的、新颖的、也是更严格的方法。使用这种方法，布局工具不得不根据设计文件中的时序约束来对布局进行评估。

模块布局工具通常具有可用的图形用户界面，并且一般很容易安装，模块级布局工具对提高版图设计的生产率是必要的。

正如下节所述，布线方法要么是基于通道的，要么是基于区域的。通道是单元间的布线区域，基于通道的布线工具利用通道来进行布线，而基于区域的布线工具则利用所有的可用区域来进行布线。

在具体的设计中，所使用的布局工具必须和布线工具紧密配合使用，因此必须熟悉布线工具的约束条件。例如，若使用布局工具对通道进行优化，那么就不应将整个设计交给Maze布线工具。

使用基于通道的布局工具，首先要明确用于布线的有效面积，其次可以扩大或压缩布线通道以适应并优化通道大小。单元被成行放置，对于一个特定的单元，它被放置在哪一行是由可布线性的最佳位置，及其他如时序、功耗等约束条件决定的。这种布局算法可以通过改变行数和长度来完成满足所有布线约束的设计。

基于Maze的布局工具假定全部在单元上方布线，因此只要在固定区域内将所有单元放置成行即可，不用考虑使用指定的布线通道。在诸多其他因素中，布局算法要重点考虑可布线性、拥塞和时序等问题。

模块布线工具。顾名思义，布线工具能自动完成一个已布局好了的设计中的连线，即完成单元与模块或芯片接口之间的连线。对于各种层次的版图设计，无论采用哪种方法，一般都可使用布线工具。

模块级布线工具具有如下特性：

- 布线工具可以自动完成数百万信号的连接，同时还可以对面积、时序、功耗等进行优化，这个能力是人工技术所无法取代的。
- 如今，要有效地利用布线工具和布局工具，需要大量的“经验”，以便充分利用内置在布线工具中的许多特性。布线工具若是不用定制的，它就是无效的，对此大多数公司已请专家来专门研究这一问题。
- 一些设计工程师不愿将布线控制权交给自动化工具，然而，人们对参数提取方法学的信心通常足以缓解这种忧虑，而且其布线速度很难被超过。对于诸如模拟电路、RF及存储器等非传统应用领域，布线技术是十分有用的，而且很可能是市场驱动力促进了布线工具在这些领域中应用。

下面将简要介绍布线工具的发展历史。了解这些工具的发展演化过程将十分有用，因为可以使我们洞悉隐含在布局布线工具背后的许多概念。

布线工具最初只是为了辅助针对当时工艺的已有模块版图设计方法，或使其实现自动化。因为当时的工艺只有一两层金属可以用来布线，因此，布线工具和版图设计工程师一样不得不在同样的约束条件下工作。

布线通道曾经运用得非常广泛。第一批布线工具拥有专门以通道布线为目标的布线算法，现今称为通道布线工具。这些工具基本上在同一方向，也就是通道的高度上优化布线。

随着可用的布线层越来越多，不同算法也相继被开发出来以充分利用这些附加分层。等到出现三到四层可用布线层时，布线通道就成为多余的东西了，一则因为能够直接在单元上方完成布线，二则因为邻近单元之间放置紧密，可以不消耗它们之间的间距。在这种情况下，Maze布线工具成为主要布线工具，因为这种工具将Maze算法内置于布线工具中，它可以基于网格或布线资源的二维区域（Maze）对布线进行优化。

近年来，基于图形的布线工具开始出现，用来处理芯片集成或者晶体管级设计。基于图形的布线工具在完成定制布线方面的功能是非常强大的，但与Maze工具以及通道布线工具相比，它的处理能力又是非常有限的。通道布线工具和Maze布线工具能够处理庞大的数据库，因为它们基本上在点对点的基础上工作，并且可以使用定义好的粗栅格。基于图形的布线工具要完成多边形，则需处理更多的信息。基于图形的布线工具将在10.3.3节中详细讲述。

在模块布线方面，与我们的讨论最为相关的是通道布线工具和Maze布线工具。

在第5章中，我们讨论了标准单元的结构是如何定义的，以及布线是怎样完成的，并且给出了一个具体的利用通道布线的版图实例。那么，通道布线工具在此环境下是如何工作的呢？

最初，在仅有一两层布线层时，第一批布线工具仅用于连接单元边界之间的信号。为了使实际通道内的信号数减少，会在通道内对布线进行优化。当可用布线层达到三层时，通道布线工具的功能增强了，它们能够在单元上方布线，并可利用单元端口在一行的中间这一特征。

通道布线工具可以优化不同布线通道的大小并将其作为初始分析步骤，它需要特定的算法来使这个分析更精确。该分析确保通道布线工具总是能够完成布线。但可能的问题是，模块完成时所占用的面积相对于原本分配给它们的面积来说可能太大了。

通道布线工具通过了解布线的结构及其限制条件来自动生成布线通道的版图。这些工具提供了如下的附加特性：

- 在连线中自动增加jog结构来减少模块的总体尺寸。
- 通过将跳线优化到合适的分层来自动减少通孔的数量。当在两个模块间布线，而它们之间只有总线而没有其他通过的信号时，这个特性非常有用。它还有助于减小芯片面积和信号线电阻。
- 可以依据特定的公式增加通孔的布局，无需局限于简单的填充式算法。
- 对于符号通道布线工具，某一信号从一个通道到另一个通道的手工转换是很容易的，同时可以运行压缩来修正DRC错误。
- 时序驱动的布局和功耗分析（如今大多数布线工具都提供这些功能）。

- 有些布线工具，它们的内置程序可以交互地调整电源连接、锥化、布线的宽度和位置等。对于一些小的几何图形来说，延迟和功耗的计算是必要的。
- 还有一些布线工具，时序分析内置在工具中，可以在布线之前或之后完成，以便设计工程师在完成最终布线前能够调整某些布线参数。时序通常是建立在引脚到引脚的延迟基础上的。
- 电流计算内置在一些工具中，它可以提供许多数据，如连线的绝对电流、电流密度、每个节点和单元的电源电压等。
- 大多数布线工具都有ECO能力。然而，要记住的是，一个复杂的修改可能会增加模块/芯片的尺寸！

通道布线也有许多不足之处。最大的不足是直到完成所有的布线，模块或芯片的尺寸都是不确定的。其次，它们将一些约束强加给标准单元库的设计。例如，单元的引脚可能被要求排列在一行上，或者被放置在单元的边界上。

正如前面提到的，当人们开始用四层金属布线来制造芯片时，通道布线工具就显得效率很低了，这时Maze布线工具开始成熟起来，同时ASIC流程中的参数提取步骤成为了标准步骤。利用Maze布线工具，设计可以是受限制的单元，并且不再需要布线通道，这就使得对芯片尺寸的估计变得更加容易，从而在整个流程中能够更早确定模块或设计的尺寸。

Maze布线工具并不处理通道，而是试图利用基于区域的方法完成布线。采用这种方法，可以依据水平或垂直的布线资源优化连接，但必须预先确定可用区域，并给布线工具加约束以确保信号线处在确定区域内。

Maze布线工具使用了不同的算法。一般情况下，使用全局布线算法来细分整个区域，使用精细的布线工具来完成信号布线，并辅以时钟树布线算法和针对电源的功耗布线算法，所有这些算法都被联系在一起来完成最终设计。

与通道布线工具每次都要完成所有布线不同，不能保证Maze布线工具每次都能完成所有线网的布线，布线拥塞和/或不可能的时序约束都可能阻止布线工作的完成，而处理这些问题的最好办法是修正单元的布局。有时，对于最后10到100个线网，利用人工调节则可能能够完成布线工作。

解决可布线性问题的最佳途径就是改变单元的布局，这就是为什么布线工具和布局工具需要紧密配合使用的原因。在这种情况下，布局工具可以对对象进行布局，这些对象是易于布线工具进行布线的，并且这只有在布局工具了解布线工具算法的情况下才是可能的。因此，我们推荐使用同一厂家的布线工具和布局工具。

Maze布线工具中的ECO功能是有用的，它可以在设计中完成较少的修改。例如，能够局部删除现有的线网，然后根据修改过的设计对其进行重新连接。

从世界范围来看，大多数ASIC设计公司运用Maze布线工具不仅是为了利用标准单元和门阵列，更是为了FPGA设计。

当完成模块布线并用提取的寄生参数进行仿真后，通常就会插入时钟树。如果知道负载分布，布局往往就是插入时钟树的起始点。在这种情况下，布线工具会用另一种算法来寻找那些连接到特定时钟上的单元，同时仅对缓冲器和时钟信号进行布局布线以使时钟相位偏差最小化，见图 10.6。

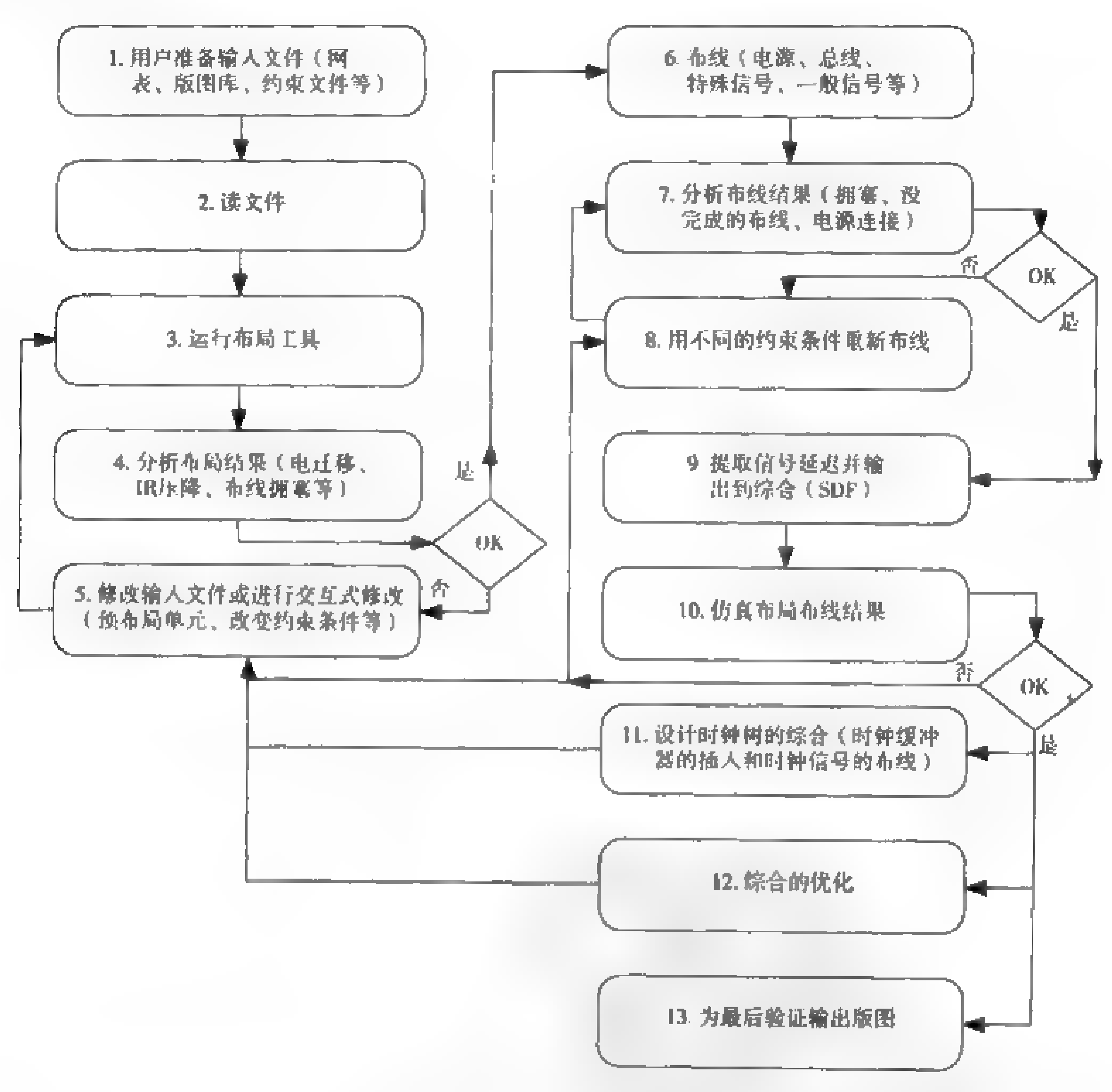


图 10.6　ASIC 布局布线流程示例

布局布线质量的衡量标准之一就是所谓的“利用率”。设计的“利用率”被定义为单元所消耗的面积与总的可用面积的比率。由于会有布线通道的面积开销，因此基于通道的设计具有较低的利用率。门阵列设计通过限制单元布局点来限制单元密度，因此，利用率约为50%~70%。

而使用Maze布线工具，其单元布局更灵活，而且可以在单元上方布线，因此利用率最高。尽管如此，由于单元被成行放置，并不是每行都能100%地布满单元，因此利用率也不能到达100%。

要记住的是，与Maze布线工具一起配合使用的布局工具使用的是固定芯片尺寸，而且不包含压缩步骤，因此，当完成所有布线后，将没有任何办法来提高设计效率。

如果能将通道布线工具中使用的方法（即先进行布线分析，再布局，后压缩）与Maze布线工具的有效布线算法结合起来，那就会有最好的布线效果了。

其思路如下：首先，充分估计设计的最佳可布线性，在此基础上，完成基于通道的布局，然后，压缩通道或行来确定最终的模块尺寸，最后，Maze布线工具利用这一最终模块尺寸来完成布线。由于基于通道的布局工具可以自行决定行的数量和长度，而提供给基于Maze的布局工具的有效面积已定，不可能通过改变模块的长宽比来提高可布线性，因此，前者最终设计的可布线性将大大高于后者。

如果用这个已经优化了的设计尺寸，Maze布线工具将更顺利地完成布线，而且利用率也会提高，因此，现在的布局布线工具市场的供应商正试图以不同的方式来实现类似的方法。

10.3.3 芯片集成工具

顾名思义，芯片集成是将芯片的不同模块和Pad单元结合起来，并将它们集成到一个完整设计中的过程。这个过程需考虑以下情况：

- 版图设计负责人利用模块布局工具来设计一个芯片的版图规划。
- 模块引脚的标识应与布线中的一致。
- 使用布线工具来连接模块。
- 设计过程中，更新规划须满足以下条件：
  当每个模块都完成后，它的空盒将被最终的模块版图所取代。
  更新布线以反映新的引脚位置和信号宽度。

通过分析这个过程可得出这样的结论：最好能在项目开始时，以“初始”版图规划的形式来完成芯片集成。在目前有效的自动化工具的帮助下，这样的流程并非梦想，而是完全可以实现的。

一般来说，对于顶层集成，或其模块数不多于50个、引脚和信号数不多于5000个的模块间互连情况，基于图形的布线工具是最佳选择。

基于图形的布线工具功能很多，但不足之处在于容量有限。通道布线工具和Maze布线工具仅在粗栅格上用简单的线形连接引脚，并不管屏蔽层的问题。而基于图形的布线工具，其众

多功能之一是可以用多边形连接不同宽度的信号，因此，它需要处理的信息更多，这就限制了它的处理容量。

基于图形的布线工具可以生成全定制的版图。因为这种工具过去是由印刷电路板的版图工具发展而来的，而在印刷电路板设计中要求可控且详细的版图设计

大多数布线工具是自动化的，但越来越多的用户要求布线工具有交互功能。一些基于图形的布线工具已经提供了前所未有的功能：

- 可以对总线的一个信号位进行手动布线，而且很容易将它的形状复制到相关总线的所有位上。
- 可以将总线作为一组一起布线，为了减小总线相位偏差或布线面积，可以将它们以45°角布线，或者绕过拐角布线。
- 通过将信号按类分成不同组，如总线、特殊组或时钟，可以约束信号布线。
- 不仅有针对模拟信号串扰、最小电容和电阻效应的基于线网的布线约束，而且每一层都有用户自定义的布线规则。

这些布线工具第一次提供了参考信号的自动屏蔽和防止天线效应的二极管应用，因此说明了这些工具是由 PCB 工具继承而来的。

10.4 支持工具

在版图输入方面，我们已经介绍了一些基本的相关工具类型。然而，设计过程不仅仅是输入版图，还必须对照各种质量标准和制造设计规则来验证版图，同时为了节省时间和精力，版图可以从另一种工艺移植过来。一些工具可以执行所有的这些操作。本节将介绍这些工具，并对目前用于版图设计的标准数据格式进行讨论。

10.4.1 版图验证工具

图 10.7 描述在第 3 章中讨论过的版图设计步骤，加阴影的步骤与本节讨论的验证工具一致。重要的是要记住，必须对整个芯片和制造商交付的版图文件进行版图验证。

DRC/LVS/ERC 随着设计规模从 1000 个晶体管增长到 10 000 000 晶体管，对于验证工具来说，如何验证整个芯片数据库文件是一个必须解决的问题。而另一方面，伴随设计水平的提高，设计工程师和工具供应商发展了方法学，使之能够验证这些设计。

从用户的角度来看，他们对验证工具的要求和对版图输入工具以及设计工具的要求不同，能否方便地调试或校正问题、工具的容量以及运行时间才是这些工具的关键所在。总的来说，验证过程是一个反馈机制，它帮助设计工程师验证版图、弄清问题和缺陷。过去的版图验证工具在解决如表 10.1 所总结的关键问题上具有局限性。

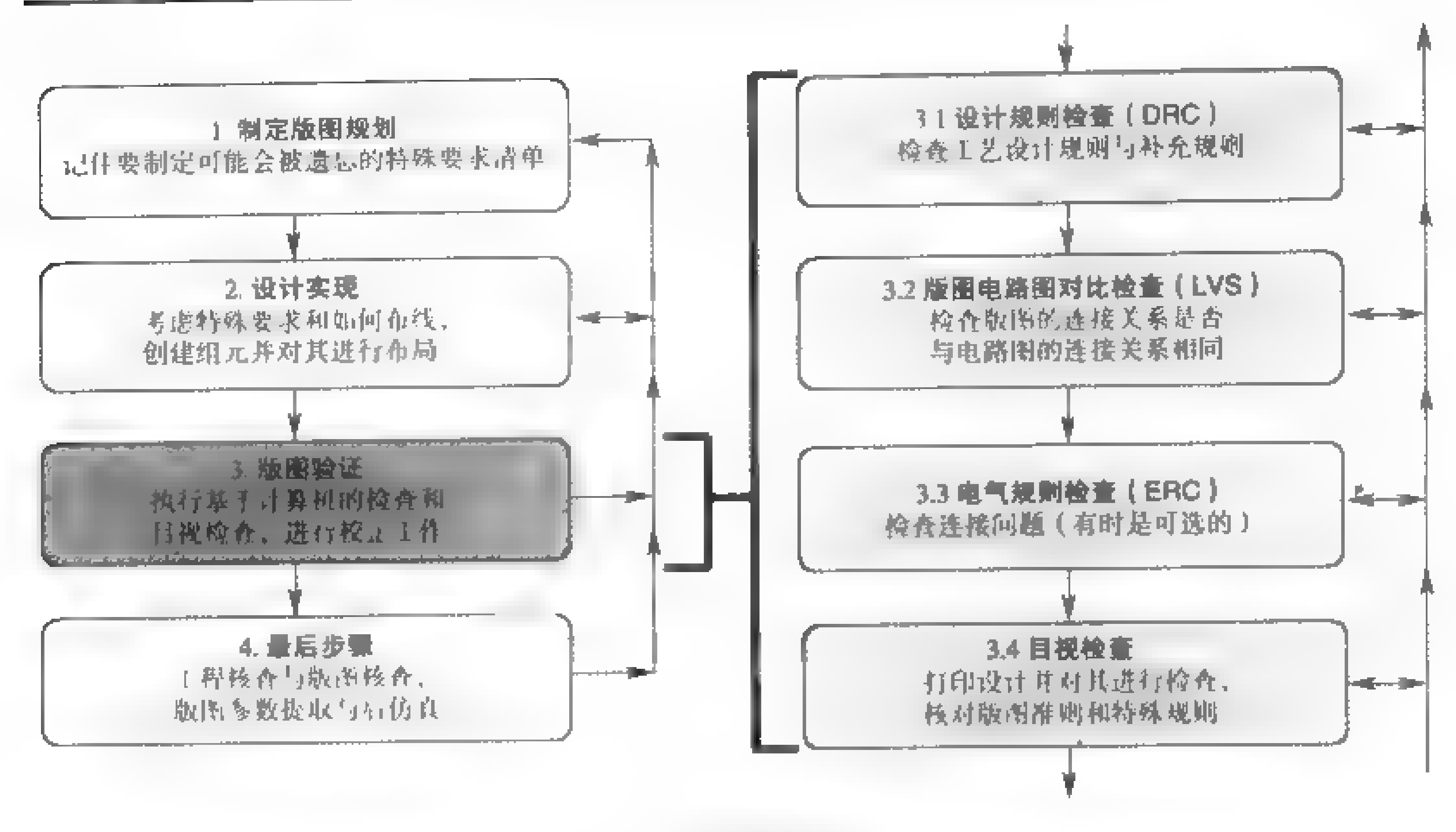


图 10.7 版图验证工具

表 10.1 验证工具的问题

关键问题	理想方法	以往的方法
用户界面	具有可视的电路设计链接和修正建议的图形调试环境	第一批工具只产生文本文件输出，但现在在 polygon pusher 工具里的交互式调试已成为标准
容量	不限制	实际限制在大约 1M 个晶体管，因此使用了“work”结构
运行时间	小于 1 小时，以至于重复迭代是可能的	花费几天还是几周的时间取决于验证的类型

无论怎样，验证的精度依赖于这些工具所采用的数值和算法。数值和算法从设置文件、命令文件和规则集合里可以获得。

“Work”结构指的是把整个芯片版图划分为更小的结构来进行验证，以确保所有潜在的问题能被发现的一种用于验证的方法。这个方法非常耗时，可能很容易出错，然而这种方法确是必要的，因为从验证工具的容量和速度上看，它们还不能一次处理一个完整芯片。

随着时代的发展，这些工具的用户界面有了很大的改进，已经十分接近我们所提到的理想用户界面。例如，DRC 错误栏是标准化的，它们能够显示违反特定规则的多边形或边缘的位置。对于 LVS 的调试来说，版图和电路图间的对照检查也是标准。如今的调试工具的特性包括基于错误类型、基于分层、基于层的分组，基于首先转到关键错误等几种方式来查找错误。

许多polygon pusher工具有"在线"版本的验证工具，能交互地执行一些小的验证工作，还能大大减小向专门工具输出数据库的时间。对于小的模块，工具的容量和运行时间的问题实际上几乎被消除了。

但必须用单独的验证工具对即将投入生产的版图进行最后验证，这些工具要能读入交付（tape-out）文件。这在过去是不可能的，因为那时验证工具的容量和运行时间还不能满足对整个版图设计进行检查的需求。

最近，人们引入了"层次化"版图验证方法来解决容量和运行时间的问题。过去，工具要载入和处理的数据量决定了它的局限性，数据量是以多边形为形式的设计尺寸的函数。

过去的验证方法忽略设计的层次结构，并且在验证过程中基本上除去设计中的所有层次结构，从而创建一个"展平式"版图数据库。这种方法确保能够将顶层单元中的多边形和单元内部的多边形一道进行检查，因此这些验证工具就需要存储和处理整个数据库。

现在的层次化版图验证采用一种完全不同的方法，它利用了设计中的层次结构。单元仅被重复地检查一次，就被置于一边，然后检查设计中余下的部分。但是，使用该工具需要对在单元上方布线的结果进行精心处理。

注意，通过对设计建立有效的层次结构，版图设计工程师可以充分利用层次化验证工具的特性。特别是，对单元上方布线的限制，使版图层次结构和电路图层次结构相匹配，是加速版图验证过程的好方法。

有关层次化版图验证环境中的关键问题将在表格10.2中再做分析。从中可以看出，版图验证的问题已经被完全解决了！

表10.2　层次化验证工具的问题

关键问题	"展平式"或过去的方法	层次化方法
用户界面	报告每个错误，不管是否重复	在重复单元中发生的错误只被报告一次
容量	实际限制为大约1M个晶体管，因此使用"work"结构	没有限制，尤其使用重复结构和层次结构时。不需要"work"结构。对交付（tape-out）文件的验证是可能的
运行时间	花费几天还是几周的时间取决于验证的类型	所有情况只需花费数小时

LVS调试工具能帮助改善电源-电源短路的定位和调试。这一错误是非常普遍的，如果不加控制，将产生许多具体的错误。

参数提取。参数提取是目前最热门的深亚微米时代的产物。版图参数提取是另一种验证版图是否实现预期行为的方法。如果DRC是检查制版规则，LVS是检查所有器件的尺寸和连接关系是否正确，那么版图参数提取则是检查在仿真中版图的性能是否能满足要求。

版图参数提取所产生的数据将版图设计结果反馈到电路设计过程中去。数据的格式可以是器件、电阻和电容的网表，或者参数提取工具也可以通过计算等效延迟或集总RC模型来简化寄生元件的网表。

对于一般的数字电路而言，最好有版图参数提取这一步骤，但是对于模拟、射频和微波设计来说则是必要的，对它们来说，一个微小的电容就能改变电路的行为特性。

由于电路设计自动化程度的提高，参数提取方法和工具随着ASIC流程的发展获得了更快速的发展，设计师不再需要用手工完成设计的各个部分了。参数提取过程给电路设计提供所需的反馈信息，并以此来评估电路的版图实现。在ASIC流程里，根据布局布线对实际版图进行参数提取的过程，对于设计的尺寸和时序是至关重要的，原因很简单：

- ASIC设计工程师不是模拟电路专家，他们只关注功能的开发。
- 设计里线网的数目，不是靠人力就可以弄清的。
- 在一般情况下，ASIC设计师甚至不需看版图。

反标是用来描述将版图信息反馈给电路设计这个步骤的一个术语。最后的仿真需要有从版图中提取的数据才能进行。为了使最终的参数提取成功，有一些基本要求：

- 版图在DRC和LVS过程中没有发现错误和警告。
- 应用精确的工艺信息对参数提取环境进行设置，并在一块小电路上进行测试，作为完整性检查。
- 以更高的精度来提取关键信号。

电路设计组应当了解参数提取的精度，这样，当他们对电路进行建模和设计时，就能解决工具的局限性了。各参数提取工具因运行时间的关系会折中考虑其精度，参见表10.3。精度上的主要区别在于参数提取工具如何计算近体效应（near-body effect）的影响。在本例中，一维的近体效应（例如，与其他信号线的耦合）根本无需考虑。三维参数求解程序不仅考虑了所有的近体效应，而且对一系列复杂方程求解来计算寄生效应的数值。

表10.3　提取工具的精确度等级

维度	描述	运行时间
一维	只在计算中使用线的长和宽	每个设计需数十分钟
二维	考虑有限的近体效应	每个设计需数十小时
准三维	基于预定义模式的寄生参数的查找表	每个设计需数十天
三维	使用三维参数求解程序来计算寄生效应	每条线网需一天

理想情况下，参数提取流程非常快而且是全自动化的，也就没有必要观察版图了。但在这种情况下，流程的设置是非常重要的。

注意，对许多专门应用来说，目视检查仍然是必须完成的。通过观察来分析不同版图结构的效果仍然是有用的。如今惟一被证明有效的方法是将单元/模块用绘图仪打印出来，然后找专家来检查版图。

绘图和绘图仪。现在，市场上绘图软件的种类还不是很多，版图设计工程师常用的有两种绘图软件：一种是简单的打印软件，使用专门打印机的驱动程序来打印版图；另一种昂贵但被广泛使用的软件是专门为“绘图”VLSI而编写的软件。让我们看看这种软件的优点：

- 用户可以从大单元中打印部分或选定的“窗口”
- 用户可以选择所要打印的特定层
- 用户可以定义与屏幕上显示有所不同的填充模式。设计工程师一般使用黑色作为屏幕工作背景，而用白色来作为打印背景。然而在黑色背景下看的清楚的图形，在白色背景下可能看不清楚，尤其是3到10层金属相互叠加的时候。
- 用户可以定义打印的缩放比例，比如1000×、5000×、10 000×，从而可以放大图形以分析模拟问题。
- 用户可以选择打印单元阵列的各种选项，例如，只打印圆环形状或者拐角。

我们为什么需要打印版图？首先是因为可以不必对DRC验证的所有规则进行编码。其次是因为在某些情况下，一些规则是非常少见的，从而DRC规则检查可能不具备确定性。第三是因为只有对大型打印图形进行目视检查，才能熟悉版图层次结构的改进。

就绘图仪而言，现在有两类应用于VLSI版图打印：静电方式或喷墨方式。静电绘图仪需要在高湿度的气候可控的房间里工作，喷墨绘图仪在室温下即可工作，因此越来越受欢迎。然而两种绘图仪各有优缺点：

- 静电绘图仪能传输打印图形长达10米。
- 喷墨绘图仪对于打印的长度有限制。
- 静电绘图仪的价格较贵，但长期来看，就打印面积的每平方米价格而言，两种绘图仪的成本是相当的，因为喷墨绘图仪的墨水和纸的价格更贵。
- 两种绘图仪的打印宽度是差不多的，因为他们都能在36、44、54英尺[1]宽的纸上绘制。
- 静电绘图仪的墨粉是以加仑为单位购买的，而喷墨绘图仪的墨水每瓶容量仅为1.36公升。
- 两种绘图仪都与网络相连，并且将打印任务按优先级排成队列。

总之，如果不需要彩色绘图仪打印图形，以完成检查图形的任务，那么根本就不需要绘图仪。黑白绘图仪多数用于掩模/光刻的检查，尽管有64英尺宽，但是仅能对照检查两层版图。当要检查四层多晶以及两到三层金属时，例如采用DRAM工艺，彩色绘图仪是必需的。

10.4.2 移植工具

在以下三种情况中使用移植工具最为有效：

① 1英尺＝0.3048米。——编者注

1. 为额外的电容或可靠电源提供第二种来源。
2. 设计重用。
3. 制造工艺的发展。

近来，硅知识产权已经成为VLSI产业中的“热门”话题，同时，版图转换工具也已经真正开始得到全面的重视。现在有两种方法来处理工艺移植：一种是在能完成工艺重定向的工具里设计版图，另一种是在设计经过硅原型验证后使用GDSII转换工具。

可以使用转换工具来改变差不多所有版图的复杂程度。过去，这种工具用来移植低层次单元和/或完整的标准单元库，它的处理速度很快且高效，但它处理的每个单元被限制在10至40个晶体管，并且VLSI工艺所需的所有分层都必须被转换。因此，当时对于这类工具，详尽的、专业的设置和维护是必要的。

为了将整个芯片从一种工艺移植到另一种工艺，转换工具提供了最好的解决方案。目前所使用的转换工具不是由电路图或网表驱动的。但是，使用表格或换算系数也能实现一些晶体管尺寸的再调整。一般来说，单元布局、引脚的位置与分配、电迁移以及RC延迟要求是一直不变的。

这些工具主要用于以下用途：

- 标准单元库的移植，在这里，引脚分配、单元高度、相邻要求等的标准化是版图设计的一个重要因素。
- 用于数据通道的单元，在这里，工具和设计要求必须得到保证并适应特定的设计。
- 整个芯片的转换。转换工具开始分层转换，源数据的大小将不再是问题。
- 如果芯片设计已处于版图设计的晚期，而此时影响芯片尺寸的工艺设计规则却发生了改变，则采用层次化模式运行转换工具几小时就能解决问题，并且完全没有DRC错误。

转换工具的优缺点如下所述：

- 这类工具使得用户可以快速移植特定类型的版图，但对于仅采用一种定义好且经过验证的工艺的公司来说，它们是非常昂贵的。新兴设计公司将极有可能投资这类转换工具，从而资助更多的此类工具的开发。
- 用户虽只需接受宏语言方面的少量培训，但是掌握高级的版图和设计知识却很关键。另外，掌握源版图的关键特性也很重要，它可以确保保持目标版图的质量。
- 安装这类工具，并将它同设计流程中相关的其他工具进行连接需要花费一些时间。
- 相对于其他任何用手工进行的版图移植而言，它们是最快的。这就是它们在过去五年中占据大量市场份额的原因。
- 缺点就是这些工具不能再添加分层，例如，将两层金属的芯片移植到三层金属的芯片。
- 它们完全尊重原来的布局；因此，它们不能利用目标工艺中可能产生的更新、更好的结构。就单元库来说，最简单的方案就是花费最小的精力去改变原设计，然后再去运行

转换。如果是整个芯片，供应商已经开发出各种层次的转换工具，例如单元的转换，单元和布线的转换，或仅布线的转换。

- 我们应该根据性能来选择工具，但也要考虑用户界面和安装的简化。即使转换是高效的，仍需花费大量的时间去安装，去调试约束，这也会增加所花费的精力。
- 有趣的是，有些硅编译工具厂商完全忽略了转换工具市场，开始致力于为其工具提供 GDSII 输入输出，以求在这块“热门”市场上占据一定的份额。

10.4.3 数据格式

任何一个设计工程师如果想用独立工具而不是在一个框架内的集成工具，就必须处理数据格式转换的问题。任何一个版图工具都始于不同的设计思想，都有不同的用途，所以它们可能会有适合自己需要的有效的数据库格式。

例如，当一个版图设计工程师想要将标准单元库中的数据传送给另一种布局布线工具时，问题就会产生。在最初的IC设计产业中，市场上只有一家公司给整个市场提供版图设计工具，那时的市场和现在比起来真是太小了，而数据格式由哪些工具来定义是受当时硬件和软件限制的。然而，后来任何想进入 VLSI 版图市场的人都需遵守这个数据格式；否则没有人会购买他们的工具。

格式从过去到现在一直是GDSII，它是由Calma公司在Data General机器上开发的。所以，如果现在你想将数据从Mentor平台输出到Cadence平台，惟一有效的方式是使用GDSII格式。当然，还有其他一些被广泛使用的数据格式，例如CIF、LEF和DEF，它们的普及也出于相似的原因：开发这些格式的 Cadence 公司占据了 IC 版图市场的最大份额。GDSII 格式是主导格式，所以有必要讨论一下。

GDSII 是一种二进制格式，从用户角度来看，具有如下性质：

- 每个流式文件只能有64层，每层可被细分为64个数据类型。所以每个流只能分为64 × 64 = 4096 个不同层来定义用于制造的多边形。
- 每个多边形和线形不能超过 199 个顶点，所以如果版图中某个多边形的顶点超过 199，那么输出子程序会将它分成只有199个顶点的小块。这是由于Calma软件对于每个多边形只能够处理 199 个坐标！
- 多边形上没有任何逻辑和电学信息。没有引脚、端口、线网或信号的识别，这对布局布线工具来说是一个很大的难题。然而，虽然没有引脚，可是有简化的方式来识别端口。编写 GDSII 文件时，从属于输出命令文件中指定层上的小多边形的端口变成了文本形式。当将这个GDSII文件输入到其他工具时，用户通常编写宏命令来选择这些文本并重新生成端口。问题在于没办法解决线网信息的保存问题。

- 将器件生成工具产生的结果、通孔和接触孔或"非多边形层次"的自动生成的版图或者软器件展平成多边形。当晶体管尺寸改变，或者数据从使用这些特性的工具中产生时，严重的问题就出现了。这是由于以往Calma公司并没有器件生成能力。
- GDSII可以识别对象的整个层次结构，但是它总是采用设计中所找到的第一个调用单元而忽略其全路径。GDSII给每一个单元单独命名，但是不能识别全路径名，这由于历史原因造成的。在UNIX系统中命名惟一意味着在全路径名中一些部分会不同。在Calma时代，单元名附属于单元库，单元库在磁盘上的位置是惟一的。

另一种布局和布线通常使用的格式是LEF，它包含了单元库和布线设置所需的版图信息，并和DEF文件一起，充分描述了线网、引脚、端口和信号的特性。

附录A 核查清单

在过去的10年里，IC设计工具在努力适应现代工艺、设计流程和设计方法发展新趋势的同时，已经获得了长足的进步。但至今仍然有少许领域，利用目前的CAD工具还不能满足其设计和工艺的要求。

为了弥补这些差距，仍需要人工来进行目视检查或技术核查。随着时代的发展，尽管CAD工具越来越完善，需要核查的项目相对减少了，但由于核查任务相当复杂，只有有经验的设计工程师才能有效地做好这份工作。

实现一个有效核查的秘诀在于，使用一份具体的、详细的核查清单，指定一位能够理解问题所在，并能提出问题解决方案的核查工程师来完成核查任务。一般情况下，核查工程师不应该是直接参与该项目的人，从而避免偏袒原有的设计风格和设计方法。

在本附录中，我们通过提供在芯片版图设计中与各种复杂程度相关的核查清单，来帮助新老设计工程师进行核查工作。以核查清单为指南，利用打印出来的版图以及所有相关的版图文档，核查工程师就可以评估版图的质量了。

表A.1列出了能应用于所有版图的通用核查清单。

表A.1 通用的版图核查清单

#	问题	答案
1.	单元是否通过DRC验证?	□是□否□不适用
2.	是否仍有部分DRC规则需通过目视检查?	□是□否□不适用
3.	单元是否通过LVS检查?	□是□否□不适用
4.	是否有一些LVS规则需通过目视检查?	□是□否□不适用
5.	单元是否通过ERC?	□是□否□不适用
6.	是否有一些ERC规则需通过目视检查?	□是□否□不适用
7.	该版图设计是否有一些特殊要求?	□是□否□不适用
8.	是否考虑到电路的关键路径?	□是□否□不适用
9.	是否尽可能地遵循版图设计准则?	□是□否□不适用
10.	是否满足了电迁移规则?	□是□否□不适用

A.1 单元

单元版图的核查清单包括以下内容，涉及晶体管版图设计问题，用于模块级设计的单元的设计。

表A.2给出了晶体管级版图设计的通用核查清单。

表 A.2 单元版图的核查清单

#	问题	答案
1.	单元是否被设计为最小尺寸?	□是□否□不适用
2.	单元是否遵循标准模板?	□是□否□不适用
3.	在单元的某些地方电源线是否被开槽?	□是□否□不适用
4.	是否所有的多晶硅线都尽可能地短?	□是□否□不适用
5.	晶体管的源/漏区是否有足够的接触孔?	□是□否□不适用
6.	晶体管的指状结构是否理想?	□是□否□不适用
7.	是否有足够的衬底和阱(tub)接触?	□是□否□不适用
8.	是否有软连接节点?	□是□否□不适用
9.	是否根据项目标准对所有端口进行了正确地配置?	□是□否□不适用
10.	单元接口设计能否确保正确的连接性?	□是□否□不适用
11.	原点是否在左下角?	□是□否□不适用

A.2 模块

模块版图核查清单需要考虑一些更全局化的问题，如模块的类型以及单元之间的连接关系。不同的模块，检查要求也不相同。例如，全定制模块、标准门或单元、数据通道、寄存器组，或乘法器模块要求对模块的功能进行特殊的检查。表 A.3 中给出了模块版图核查清单，它针对多数情况下极为重要的通用议题概括出需要核查的问题。

表 A.3 模块版图核查清单

#	问题	答案
1.	模块是否遵循版图规划?	□是□否□不适用
2.	是否根据对功耗、电迁移和 RC 参数要求的仿真结果来确定电源网格?	□是□否□不适用
3.	电源条是否适量，其上是否有足够的通孔?	□是□否□不适用
4.	在模块的某些地方电源线是否被开槽?	□是□否□不适用
5.	是否优化了所有的关键路径信号线的长度?	□是□否□不适用
6.	是否已经满足所有的特殊信号要求?	□是□否□不适用
7.	是否根据项目标准对所有端口进行了正确地配置?	□是□否□不适用
8.	模块接口设计能否确保正确的连接性?	□是□否□不适用
9.	是否有足够的备用连线和备用逻辑?	□是□否□不适用
10.	对于特殊信号是否有探测 Pad?	□是□否□不适用
11.	原点是否在左下角?	□是□否□不适用

A.3 芯片

对于芯片级的版图核查而言，核查的复杂程度更大。在这种情况下，需要验证的问题有很多种，往往取决于设计的类型。举个例子，ASIC 的核查清单要短些，而全定制模拟芯片的核查清单则非常长。不同芯片的设计工艺和方法差异很大，核查清单不可能涵盖所需核查的所有

内容。表A.4中给出了模块版图核查清单，它针对多数情况下极为重要的通用议题概括出了需要核查的问题。

表A.4 芯片版图核查清单

#	问题	答案
1	芯片是否满足所有的封装要求?	□是□否□不适用
2.	连接到Pad的电源是否充足?	□是□否□不适用
3	电源条是否适量，其上是否有足够的通孔?	□是□否□不适用
4.	在顶层布线的某些地方电源线是否被开槽?	□是□否□不适用
5	是否优化了所有的关键路径信号线的长度?	□是□否□不适用
6.	是否满足特殊信号的所有要求?	□是□否□不适用
7.	是否有足够的备用连线和备用逻辑?	□是□否□不适用
8	对于特殊信号是否有探测Pad?	□是□否□不适用
9	敏感电路是否放置在芯片的边缘?	□是□否□不适用
10.	是否满足针对芯片拐角区域的规则?	□是□否□不适用
11.	是否正确定义了芯片和划片槽之间的接口?	□是□否□不适用
12.	是否满足了所有的ESD和Pad的闩锁效应要求?	□是□否□不适用
13.	是否包括了所有必需的芯片结尾单元?	□是□否□不适用
14.	原点是否在芯片的中心处?	□是□否□不适用

附录 B　数据库管理

在实施项目的过程中，不同组的成员共同工作，并行完成设计的不同方面。动态且快速地生成、修改、共享以及删除不同类型的数据。这些数据可能是版图，也可能是电路图、设置文件、文档以及其他类型的数据。那么，该如何管理所有这些数据呢？答案是规范的数据库管理技术。

数据库管理是一个由基础结构支持的过程，这个基础结构应具有以下特性和优点：

- **版本控制：** 每个被管理的文件都应该有一个与之相关的版本信息。
- **版本历史记录：** 所有数据库对象必须有历史记录可寻。
- **数据共享：** 数据必须可以共享，并应以系统的方式完成数据共享。对共享单元的修改不应影响正在使用这些单元的工作。
- **数据库完整性：** 保护数据以防止无意识地删除或损坏数据库。数据库损坏的例子有：在一个层次化设计中缺少被引用的单元，在计算机系统中不同的地方有两个名字相同的单元。

这里提到的最后一点，正是规范数据库管理过程的主要原因。当完成模块设计，交付日期临近时，数据库的完整性就显得至关重要。

几乎每个公司都有不同的方法来处理这个问题，然而基本要点和理念总是相似的。事实上，为了便于理解，我们通常总是将基本的数据库管理方法和版图数据管理方案一起提出。

在实际应用时，IC 设计中只有一些基础概念使得系统正常工作。数据库管理的作用就是要确保以下几点：

- 电路和版图设计之间通常存在着一一对应的关系。
- 共享的或更低层次的单元在被其他组成员使用之前是不允许修改的。
- 在任一给定时刻，只允许一个人修改某个单元。
- 当一个单元正在被修改时，其他组成员可以使用其原来的版本。
- 根据要求，应发布新的版本并就此新版本进行交流。

规范的数据库管理依赖于完善的计算机系统，该系统应该可以识别不同组的用户，以便对访问数据的人进行鉴定。根据数据的类型和应用功能，系统中数据的访问权限应该指定为三种用户组权限：

- **全体权限（World access）**：每个人——对于全体必需的数据，如CAD软件。
- **小组权限（Group access）**：小组成员——受限并且需身份认证的数据只面向一个项目组。
- **所有者权限（Owner access）**：最终修改或操作数据的个人。

对数据操作的访问权限也应该加以识别：

- **写权限（Write access）**：可以修改数据。
- **读权限（Read access）**：可以使用和调用数据，但不能修改数据。

图B.1描绘了单元的创建过程。

在这个例子中，Fred是DSP32项目版图组的成员。Fred的工作是创建一个名为AGBC的单元。这时，应该注意的关键事情是：

- Fred应先检查数据库，以确保数据库中不存在这个单元名。这一点很重要，可以避免使Fred做一些无用功，因为也许这个单元版图已经被无意中分派给其他人了或已经被做完了。
- 一个单元只有当完全通过验证后，才可以"入库"（check in）。也就是说，只有在这个时候，小组才能享有读权限。"Check in"是一个术语，用来指将单元提供给小组其他人使用，类似于把书归还给图书馆。
- Fred只有在单元"出库"（check out）之后，才能够修改这个单元，这里，"check out"也是一个术语，用来指从数据库中取得单元的控制权。

我们用这种方式就可以系统地创建单元了，并且可以很容易地维护单元的历史记录。在check-in阶段，对数据库进行自动检查的功能可以内置于系统中。

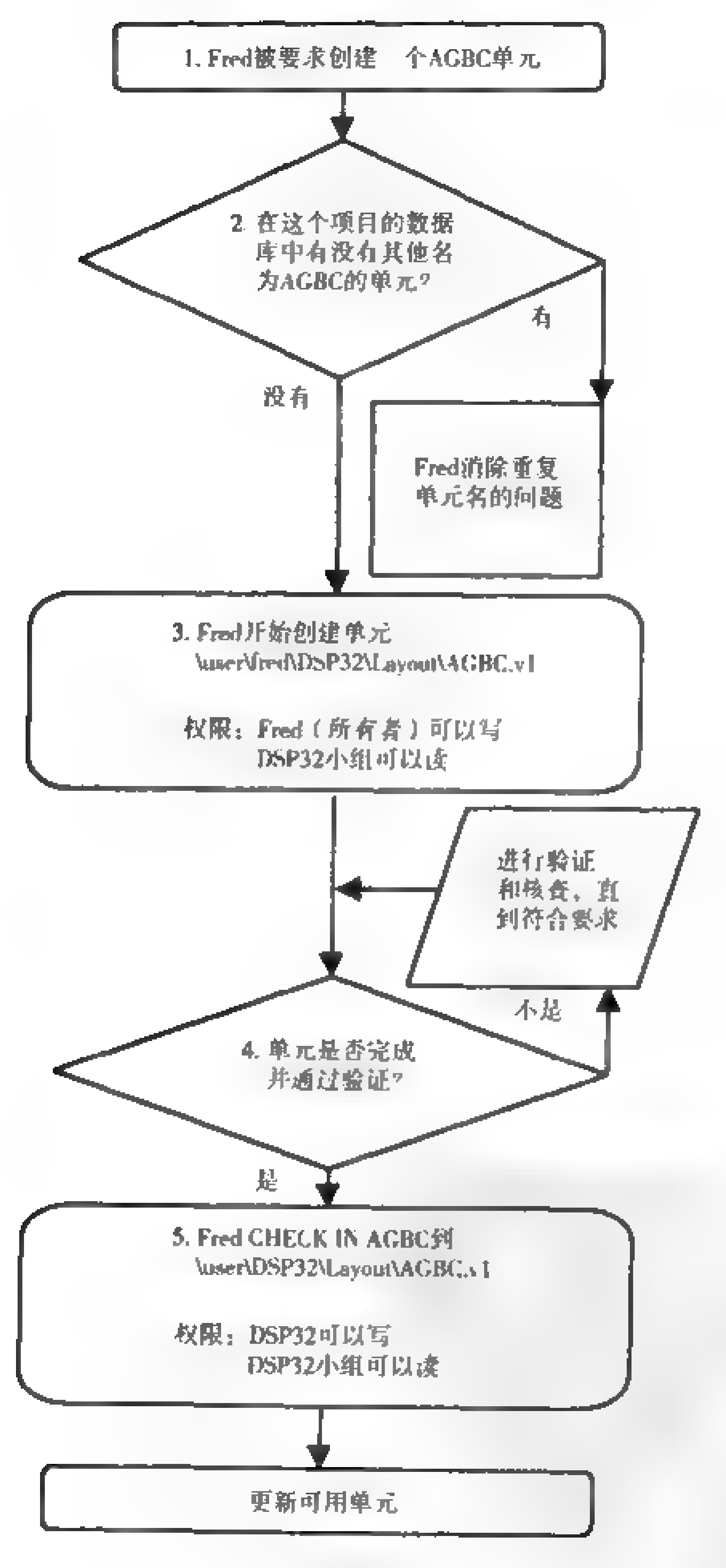
图B.2所示的例子更加有趣，因为现在要求对AGBC单元进行修改。

在这个例子中Dan、Julia和Brenda也都是DSP32项目版图组的成员。Fred这次所要做的工作是修改名为AGBC的单元。这时，应该注意的关键事情是：

- 必须通知Dan、Julia和Brenda单元将会有一些修改
- Dan、Julia和Brenda必须引用以前版本的AGBC单元，直到它的修改完成为止。
- 当AGBC新版本可以使用时，要通知Dan、Julia和Brenda，这样，在适当的时候，他们就可以更新自己的版本了。

输入到中心数据库和从数据库输出的check-in和check-out的过程，将有助于所有设计工程师保持自己的工作进度，不受他人修改数据的影响。

一旦数据库发生变化，以上提到的这些技巧就会帮助我们实现系统的数据库管理。



图B.1 示例单元创建的步骤

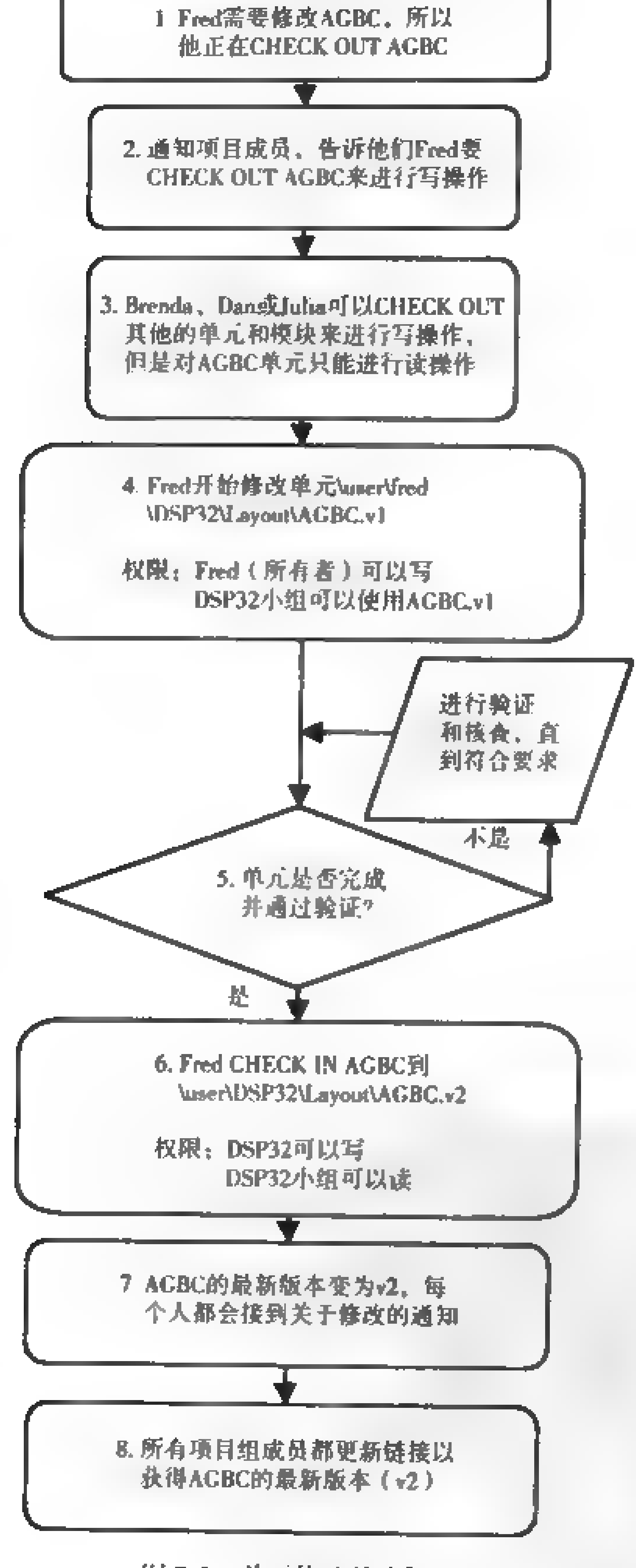


图 B.2　单元修改的步骤示例

附录C 进度安排

安排好单元、模块或者芯片的版图设计进度通常是一个经验问题。本附录既提出了版图设计产业的平均速度，又对具体的版图进度安排给出了一些建议。

预测一个项目的进度表是一项复杂的任务，它取决于很多因素：工具、流程、硬件、设计团队的经验、培训、资源、假期、病假、工艺修改等。根据我们的经验，加快进度的最好方法并不是使用更快的计算机或更好的软件，而是提高设计团队的经验。为了获得最高的生产率，没有任何东西可以替代专业技术所能起到的作用。

让我们来考虑一下如何安排一个单元版图设计所占用的时间。在所有的进度安排中，必须考虑中等水平设计工程师的速度，而不是最好的设计工程师的速度，这一点很重要，因为进度表要能反映各设计小组的真实情况。

对于单元级的进度安排，我们需要下面的信息：

- 晶体管数。
- 信号数。
- 单元类型——它是由一个模板生成的，还是一个全新的？
- 特殊要求。

表C.1给出了不同情形下完成单元版图设计所需的时间。

注意：单元类型和特殊要求确实会影响完成单元所花费的时间，而验证所用的时间并不包括在表格中，因为我们通常假定对于不同的单元，验证所需的时间是相同的。

就模块级而言，应有一些不同的考虑：

- 组元的数量——单元、小模块、随机逻辑门。
- 总线、信号和电源网格要求的数目。
- 特殊要求——对称、串扰、最小RC、时序。
- 尺寸限制。
- 有效布线分层——例如，五层布线层中只能用三层来布线。

表C.2是一份模块级的进度表。

影响整个芯片进度的因素如下：

- 团队的经验水平：团队是否有足够的有经验的成员？

- 修改：估计一下项目过程中某些主要参数修改所带来的风险，例如，Pad的位置、在一些长期的项目中，甚至设计规则也可能发生变化。
- 重用：是否能利用已有的经验和版图设计？
- 设计复杂度：关键模块、信号和/或总线的数目。
- 外在因素：该项目是否独立完成？还是与其他公司合作完成？
- 第三方的模块：从IP供应商处引入模块的难易程度。
- 方法学：是否引进新的流程和/或工具？
- CAD支持：这个项目小组是否有CAD支持？这个项目享有哪些优先权？
- 团队规模：由于交流和管理开销的原因，大型团队可能没有小型团队效率高。
- 工作日：是假设加班还是计划加班？
- 病假、丧亲假、假期、季节限制。

项目的进度安排是一门艺术，也是一门科学。前面所列的表格均说明了这项任务的复杂程度。

表C.1 以小时为单位的单元级进度表

单元名	晶体管编号	信号编号	类型	特殊要求	用手工花费的时间	自动的*
INV1	2	2	STD	无	4	0.25
INV × 20	2	2	STD	最小电容	8	0.25
DFF	16	6	STD	最小尺寸	16	1
INV1	2	2	无	无	1	0.25
INV × 20	2	2	无	无	2	0.25
DFF	16	6	无	无	8	0.5
INV1	2	2	数据通道	最小尺寸	4–8	0.5
INV × 20	2	2	数据通道	最小尺寸	4–8	1
DFF	16	6	数据通道	最小尺寸	8–24	1

* 项目进度表中还需增加安装工具和结构设置所需时间，但是用在手工设计的项目中则不需花费这些时间。
时间＝直接工作时间，不包括其他时间开销（会议、休息、午餐）
STD＝库的标准单元
数据通道＝具有很多N/P/P/N区域的乘法器、寄存器组
无＝正常的随机逻辑
注意：该进度表并不包括验证所需的时间

表C.2 以小时为单位的模块级进度表

模块名	晶体管编号	信号编号	类型	特殊要求	用手工花费的时间	自动的*
控制单元	200	150	STD	无	40	2
同步模块	400	300	STD	无	80	4
寄存器组	10 × 32	8 × 32 + 30	STD	无	80	4
控制单元	200	150	STD	Timing	60	4
同步模块	400	300	STD	Timing	100	8
寄存器组	10 × 32	8 × 32 + 30	STD	Timing	90	8

（续表）

模块名	晶体管编号	信号编号	类型	特殊要求	用手工花费的时间	自动的*
控制单元	200	150	STD	Timing + Size	70	5
同步模块	400	300	STD	Timing + Size	120	9
寄存器组	10 × 32	8 × 32 + 30	STD	Timing + Size	90	9

* 项目进度表中还需增加安装工具和结构设置所需时间，但是在用手工设计的项目中则无需花费这些时间。

时间 = 直接工作时间，不包括其他时间开销（会议、休息、午餐）。

STD = 利用标准单元生成的逻辑门（标准单元类型）。

Timing = 符合时序要求的布局和布线。

Size = 最小可能的尺寸。

无 = 正常的随机逻辑。

注意：该进度表并不包括验证所需时间。

索　引

